Aspect and Opinion Extraction for Amazon Reviews

Achyut Joshi[1], Andrew Giannotto[2], Ishika Arora[3] and Sumedha Raman[4]

***Abstract*— Opinion mining or sentiment analysis is the computational analysis of a person's emotion towards entities like products and services. It can be done at three levels - document, sentence and aspect. We have implemented an aspect-based analysis system to extract various aspects of an entity from Amazon product reviews, group them and determine the respective polarities.**

I. INTRODUCTION AND MOTIVATION

Consumers and sellers spend a large amount of time reading through long reviews to find out what is perceived as good and bad about a product. Amazon currently has a feature that lets users filter reviews by popular keywords, which is still tedious and time-consuming for customers. The users have to read through numerous reviews to get the relevant information about the products that they need. Our model adds an additional layer on amazon reviews system, which extracts key aspects of a product, groups them, and determines their polarity.

II. PROBLEM DEFINITION

The goal of our model is to identify different aspects of a product that have been reviewed, and determine the polarity of the most common aspects. To ascertain which aspects to consider, we extract adjective-noun pairs from the review text in Amazon Customer Reviews Dataset. The nouns in these pairs are used to select aspects, and the adjectives are used to determine polarity. Results from the model will be displayed through an interactive UI which shows most popular aspects of a product, grouped into positive and negative. The UI will also direct users to the specific reviews that mention the selected aspect.

III. SURVEY

A. Sentiment analysis of Amazon reviews and perception of product

The author identified adjective-noun pairs using the Stanford Dependency Parser and inferred the polarities of adjectives by considering the weights of other words in sentences and their polarities [1]. However, their results for polarity deduction were mixed. We incorporated a slightly different method for extracting pairs, and will improve upon their polarity deduction model.

B. Review Spotlight: A User Interface for Summarizing User-generated Reviews Using Adjective-Noun Word Pairs

The paper used a POS tagger to extract and display the most common adjective-noun pairs, sized according to frequency and color-coded based on the polarity[2]. However, it did not consider similarity among nouns in adjective-noun pairs, so the results had some redundancy. Our model provides a visualization that summarizes the opinions, and group nouns with similar meaning.

C. Mining the peanut gallery:opinion extraction and semantic classification of product reviews

Their approach scored words based on the average of the scalar ratings of the documents in which the word appears, and used n-grams to determine the aspects [3]. The n-gram approach is still prone to redundancy since there could be n-grams with similar meaning that are considered separate entities. Our model takes into consideration the similarities of features while grouping aspects.

D. Aspect and Entity Extraction for Opinion Mining

It divided the task into three parts- (1) identifying and extracting entities (2) identifying and extracting aspects (3) determining sentiment polarities on aspects[4]. We tested their method of determining sentiment polarities for Amazon reviews in particular.

E. Building a Sentiment Summarizer for Local Service reviews

The authors used user-provided labels as a prior knowledge of reviews and applied aspect-based sentiment analysis[5]. They implemented a hybrid aspect extraction system - dynamic and static, and summarized the reviews. Our model extends this analysis by identifying aspects dynamically without user-provided labels.

F. Feature Specific Sentiment Analysis for Product Reviews

It develops a case that associated words come together to express an opinion[6]. The authors used Stanford dependency parser to understand relationship between nouns & adjectives and group highly related entities. We also used dependency parsing to extract the pairs, but our model recognizes similarities between features as well.

G. Sentiment Analysis in Amazon Reviews Using Probabilistic Machine Learning

It uses sentiment analysis to assign polarity to Amazon reviews based on words in the text [7].We enhanced this by using sentiments of the product aspects and not just words mentioned in reviews.

H. Effect of Adjective Orientation and Gradability on Sentence Subjectivity

It focuses on determining the subjectivity of a sentence by analyzing adjectives [8]. Since it does a good job of determining subjectivity, we tested their design to improve our model while determining polarities of adjectives.

I. Feature Generation for Text Categorization Using World Knowledge

It enhances the bag-of-words model by additionally referencing large publicly available ontologies [9]. This study ultimately identifies key features/categories in the text which can be linked to aspects of a product. Our model used the same concept but groups together nouns based on word vectors.

J. Sentiment analysis of product reviews: A review

Building on analysis of unstructured text data, they compare the machine learning based and lexicon based approaches for sentiment classification[10]. We leveraged this analysis to select the most optimal method for sentiment analysis of adjective-noun pairs.

K. Aspect based Sentiment Analysis using SVM classifier

Aspect-based sentiment analysis was done on a small dataset of product reviews from e-commerce websites using SVM classifier[11][12]. We referenced these methods to define our aspect-opinion identification methodology. We worked on a larger dataset (around 5M+ reviews), and also grouping aspects into meaningful categories.

L. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text

This is a rule-based model for sentiment analysis, which has been proven more effective than most benchmarks for sentiment analysis as well as human raters [13]. It has been integrated into the NLTK library. We used this feature to calculate polarities.

IV. PROPOSED METHOD

The intuition behind our model is that the aspects extracted from a set of reviews of a product can be similar or related to one other. So considering the most common aspects verbatim for sentiment analysis might not be an accurate representation of the opinions in the reviews. Users may discuss the same features of a product in different words; clustering the aspects before determining the most frequently mentioned aspects would prevent the omission of key aspects. Additionally, it will also ensure that there is no redundancy. Once the aspects are clustered, the polarity for each cluster is calculated as the mean of the polarities of all aspects that belong to the cluster.

A. Data Collection & Hosting

The Amazon Customer Review Dataset is available for free in the Registry of Open Data on AWS. It consists of two decades of reviews from 1995 to 2015, for 50 different product categories. The complete data set consists of approximately

130M+ reviews. For testing our model code and UI, we downloaded the 'Electronics' review data set file and ran it locally. Since the file size were huge and our model requires memory and time to run, we ran the model on the complete file on AWS. The data set files were directly available as .tsv files on S3 bucket, so the code would fetch data from these files directly when running the model on AWS.

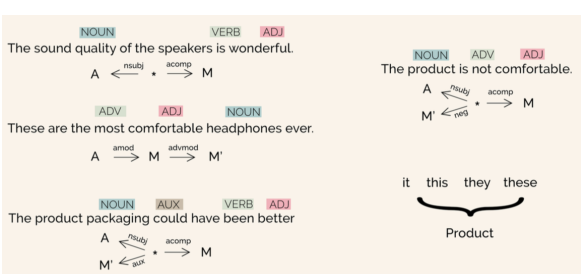
B. Data Cleaning

The review text that we extracted from the data set had a lot of unclean data, so We created a cleaning script for the dataset. It removed unnecessary characters, hyperlinks, symbols, excess spaces, and other patterns of text that could not be processed by our algorithms. From the cleaned dataset, we extracted the review text description for our analysis.

C. Extraction of Adjective-Noun Pairs

The objective of this step was to extract instances of product aspects and modifiers that express the opinion about a particular aspect. We used the dependency parser tree in Python's spaCy package to extract pairs of words based on specific syntactic dependency paths. The output of this step was a list of such noun-adjective which serve as the input to the next step of grouping aspects.

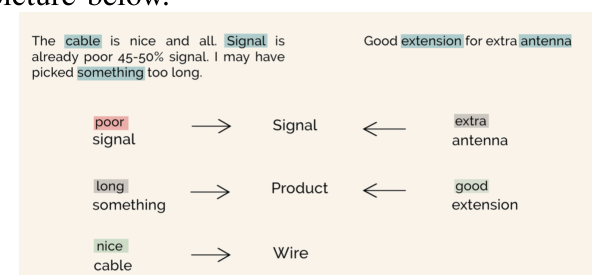
We formulated rules based on the POS tagging of the words in a review sentence. As an example, a word with "nsubj" dependency relationship with a verb token would be the noun of the phrase, and a word with "acomp" dependency relationship would be the adjective of this noun. Thus, we would extract this pair as a relevat aspect-modifier pair. Some of the rules we formulated are shown in the picture below.



D. Grouping Aspects

For grouping aspects, we determined word similarity by comparing word vectors or "word embeddings", generated using inbuilt vector model in Python's spaCy package. We used spaCy for vectorization as it provides fast and easy access to over a million unique word vectors, and its multi-task CNN model is trained on 'web' data and not 'newspaper' data as in other libraries like NLTK.

The word vectors were then grouped using K-Means clustering algorithm in Scikit-Learn. We experimented with other clustering algorithms such as DBSCAN. However, K-Means gave us optimal results with four clusters. The clusters were labeled based on the most frequently appearing word in each cluster. Some examples of how the aspect pairs were grouped into clusters are shown in the picture below.



E. Determining Polarity and Score

To determine polarity of the aspects, we used the VADER Sentiment Analysis tool which is part of the NLTK library. We chose this over other tools such as spaCy and TextBlob because of the accuracy and speed. With the set of adjectives in a cluster as input, we aggregated the polarity compound score provided by the tool using SQL queries to calculate an overall polarity for each cluster.

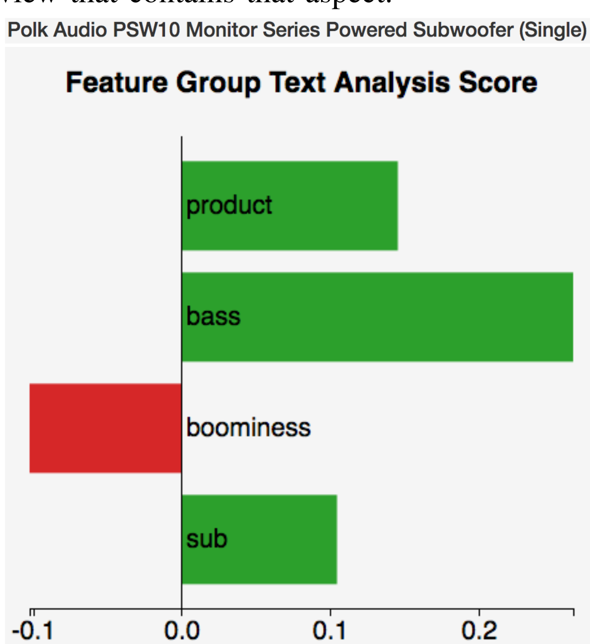
F. Database

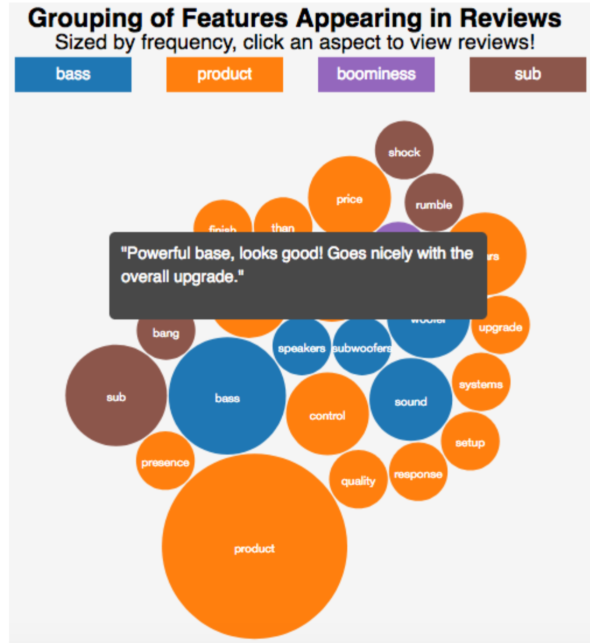
To make it easier for the UI to pull data and ensure that it's easier to keep updating the model results in future as we run it on more data, we hosted our model results on a Microsoft SQL Server database. We are hosting the database on AWS to store the product, review, aspect and cluster data, and aggregate the polarity. Based on the results obtained from the model, we imported data into the database using Sequelize. A Node.js server was set up in the AWS machine which accepts requests with the product ID, and returns the query results needed for the visualizations.

G. Framework, UI and Visualization

We implemented a Flask-based UI with search functionality for Amazon reviews based on product ID, limited to those in the dataset. The product ID can be easily fetched from the Amazon product page URL. The search page redirects the user to a product landing page which displays a detailed analysis of the aspects and polarity score results from our model. An Ajax request is sent to the database, which returns the results of our models for the searched product ID. If the product ID does not exist, a message is displayed and the user is returned to the search page. The product landing page is hosted locally for now. We used Flask, jQuery, HTML, and D3.js for development.

The product landing page displays a horizontal bar chart with clusters along with their aggregated polarity scores. The green and red bars denote an overall positive and negative opinion, respectively. We also added a bubble chart with color-coded aspects, where the bubbles are colored according to the cluster and sized according to the frequency of the aspects. Additionally, there is a tool tip that allows the users to click on a bubble and see a review that contains that aspect.





V. DESIGN OF EXPERIMENTS

We conducted experiments for each stage of the implementation: Aspect Extraction, Clustering, Polarity Calculation, and UI. For aspect extraction, the goal was to check whether we are identifying the relevant pairs of nouns and adjectives from the reviews, and that we are not extracting a lot of noisy data. The clustering experiments aimed at ensuring that similar aspects were grouped and the cluster name was chosen optimally. The polarity metric was evaluated in comparison with the star rating given by reviewers. Finally, the UI experiments intended to understand the best visualization technique to demonstrate the opinions in the reviews quickly and effectively.

VI. EVALUATIONS AND OBSERVATIONS

A. Evaluation of Extraction of Adjective-Noun Pairs

To test whether the adjective-noun pairs are extracted correctly, we verified this manually for a set of 100 sample reviews. The pairs obtained from a review were compared with the review text to check if the relation between the nouns and adjectives in a sentence were mapped correctly. This was done in iterations and the grammar rules were updated accordingly.

As an example, our initial set of rules was not treating the negations in the reviews properly.

We had to add special rules to handle negation for different types of "neg" modifier relationships in the sentence. Words like "could have been", "should have been" display a negative sentiment in the sentence, however a simple dependency rule wouldn't capture this relationship. So we added a special rule to append a "not" before the modifier for such pairs. This helped us determine the correct polarity scores for such sentences, e.g.

The product packaging could have been *better.*
— *(packaging, not better)*

B. Evaluation of Clusters

For all nouns that are grouped into a particular cluster, we tried calculating the average cosine similarity among the nouns using Word2Vec in order to name the cluster (on a smaller set of model output). Since this method proved to be computationally expensive, we used an alternative method for the dataset wherein the cluster name was decided by the most frequent aspect in the cluster.

C. Evaluation of Polarity

To evaluate the polarity, we aggregated the polarities of aspect pairs for each review. After scaling it to the range of 1-5, we compared this value to the star rating provided for the same sets of reviews. As seen in Fig. 1 for a subset of reviews, our model calculates a score within a relatively small range; there are no scores in either extremity.

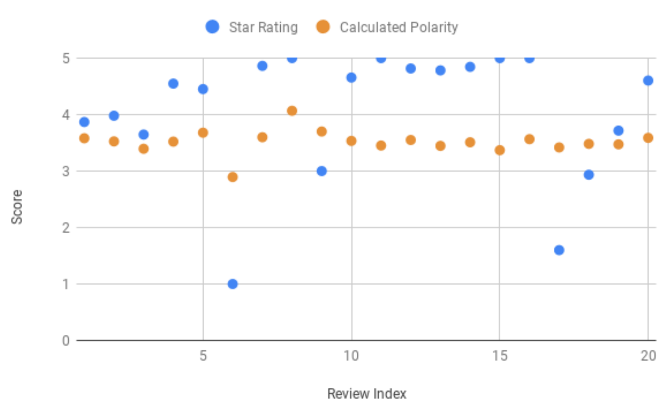


Fig. 1 Comparison of Calculated Polarity with Star Rating

Our polarity metric worked well for unambiguous positive or negative words; however, for some equivocal adjectives which require context for interpretation, the metric did not give an accurate value. For example,

"The wire is too *thick*, this makes it *bulky*"
"The blanket is so *thick* and *warm*!"

From the context, *thick* is meant to be negative in the first review but positive in the second. Since our model only considered adjective-noun pairs, it could not make this distinction and hence returned a neutral score.

D. Evaluation of Visualization Technique

The efficacy of the model's visualizations was evaluated using user surveys in order to check if the aspects and polarities give a coherent summary of the opinions in the reviews, and also whether the visualization delivers the key information faster than reading through text in reviews. Based on the iterations of feedback, we decided to use a bubble chart and bar chart.

VII. WORK DISTRIBUTION

All team members have contributed similar amount of effort. The distribution of work in the team is shown in Table I.

TABLE I

Aspect Mining	Achyut, Ishika	4 weeks
Clustering Aspects	Achyut, Ishika	4 weeks
Determining Polarity	Sumedha	2 weeks
UI, DB & Framework	Andrew, Sumedha	4 weeks
Testing & Experiments	Everyone	2 weeks

VIII. INNOVATIONS

1) Dynamic Aspect Identification: Using a static set of aspects does not scale to all categories of products, since the aspects will depend on category. Our model uses the review corpus to automatically identify the common aspects that are addressed for each product.
2) Grouping of Common Keywords: When the most common keywords are displayed, there could be related words that convey the same meaning but are phrased differently; this can lead to redundancy. Our model addresses this by grouping keywords based on their cosine similarity scores.

3) Interactive Visualization of Aspects and Polarities: The interactive visualization is a key innovation in this model. The visualization of the aspects enables users to see the overall polarity of a particular aspect in comparison with other aspects of the product. It also displays some of the relevant reviews in a tool-tip when the user hovers over a certain aspect.

IX. CONCLUSION AND DISCUSSION

The model proved to be effective in conveying opinions in Amazon reviews using aspect extraction and clustering. The polarity calculation, however, can be improved upon in future work using disambiguation and implementing a contextual approach to determine polarity scores. Apart from this, the user interface is currently a website that takes in the Amazon product ID and displays model results. This can be added as a browser plugin to eliminate the extra step required from the user's end. Also, currently the model training time is around 7-8 hrs for 1M reviews. We can further work on optimizing this to reduce the model run time.

REFERENCES

[1] A. Wallin, Sentiment analysis of Amazon reviews and perception of product features, 2014.

[2] Koji Yatani, Michael Novati, Andrew Trusty, Khai N. Truong, Review Spotlight: A User Interface for Summarizing User-Generated Reviews using Adjective-Noun Word Pairs, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, May 2011.

[3] Kushal Dave, Steve Lawrence, David M. Pennock, Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews, Proceedings of the 12th international conference on World Wide Web, May 2003.

[4] Lei Zhang Bing Liu Aspect and entity extraction for opinion mining in Data Mining and Knowledge Discovery for Big Data Springer pp. 1-40 2014

[5] Blair-Goldensohn, Sasha., Kerry, Hannan., Ryan, McDonald., Tyler, Neylon., George A. Reis, Jeff, Reyna. 2008. Building Sentiment Summarizer for Local Service Reviews In Proceedings of the Workshop of NLPIX . WWW, 2008

[6] Mukherjee S., Bhattacharyya P. (2012) Feature Specific Sentiment Analysis for Product Reviews. In: Gelbukh A. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2012. Lecture Notes in Computer Science, vol 7181. Springer, Berlin, Heidelberg

[7] Rain, C. (2013). Sentiment Analysis in Amazon Reviews Using Probabilistic Machine Learning. Swarthmore College.

[8] Hatzivassiloglou, V., & Wiebe, J. M. (2000, July). Effects of adjective orientation and gradability on sentence subjectivity. In Proceedings of the 18th conference on Computational linguistics-Volume 1 (pp. 299-305). Association for Computational Linguistics.

[9] Gabrilovich, E., & Markovitch, S. (2005, July). Feature generation for text categorization using world knowledge. In IJCAI (Vol. 5, pp. 1048-1053).

[10] JT. K. Shivaprasad and J. Shetty. Sentiment analysis of product reviews: A review. 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2017, pp. 298-301.

[11] R. Varghese and M. Jayasree. Aspect based Sentiment Analysis using support vector machine classifier. 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Mysore, 2013, pp. 1581-1586.

[12] D. V. N. Devi, C. K. Kumar and S. Prasad. A Feature Based Approach for Sentiment Analysis by Using Support Vector Machine. 2016 IEEE 6th International Conference on Advanced Computing (IACC), Bhimavaram, 2016, pp. 3-8.

[13] Hutto, C.J. Gilbert, E.E. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.