Austin and Cedric

Big Data Report unformatted

**Introduction**

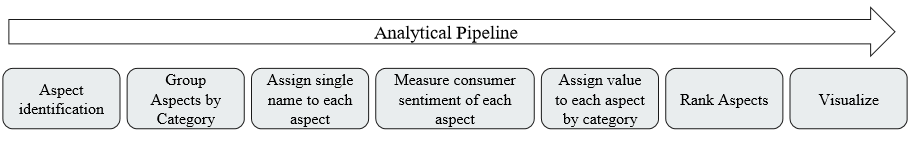
Why does one customer choose to purchase one item over another? If two items are truly substitutes (i.e., they perform the same function), there must be some defining features or aspects that lead to purchase of specific products. These are common questions asked across all industries and even across microeconomic academic circles.

Traditionally, when trying to identify the causal factors of a consumer’s purchase two methods are typically utilized:

1. Experiments with a control group and treatment group
2. Surveying customers followed by a conjoint analysis

Both of which come with strong disadvantages. Traditional experiments are incredibly costly and time consuming outside of the software industry (i.e., AB testing), whereas survey analysis introduces strong bias where a participant may answer in the form that they believe the surveyor wants them to.

Our research develops a complete methodology for understanding the relative importance of a product’s aspects within the retail sector. Our method leverages the unstructured text in product reviews to identify aspects, clustering algorithms to select the common aspect phrases, sentiment analysis to determine the user’s opinion towards the aspect, and then uses a probabilistic ranking model to evaluate each feature’s importance. We apply our model across a wide variety of unrelated product categories to demonstrate the generalizability of our model across the retail sector. Specifically, our model leverages the following pipeline:



*Figure 1:General Analytical Pipeline*

The result is that for each product, we will have identified the key product aspects and their relative importance. Since we identify aspects at the category level, we then are able to easily compare aspects and aspect importance over a wide range of products.

We must acknowledge that this is just the first step in causal analysis to understanding consumer reasons for purchases, but we must impart the two key advantages over traditional approaches to gathering such data:

1. The analysis time after model development is extremely quick, representing a significant improvement compared to experiments or surveys
2. Potential surveyee bias has effectively been removed by leveraging open-ended unstructured reviews volunteered by purchasers of the product without knowledge of this research

**Relevant Work**

Due to the multi-layered approach of our work, there exists multiple bodies of literature that overlap with our analysis including aspect extraction, sentiment analysis, and aspect ranking. We provide a basic overview of some of the work from each of these groups that have influenced our analysis.

Aspect Extraction:

The first portion of our framework requires identifying and extracting the aspects. Much of the past research have extracted aspects based on the frequency of nouns or noun chunks. For each review or product, word counts and TF-IDF weightings are calculated and the top *n* aspects are extracted. While this results in extracting all possible aspects, this methodology results in a significant amount of noise. Past research has typically accounted for this additional noise through leveraging data external to the unstructured product reviews such as leveraging a Pros and Cons input (Zha et al, 2014), through leveraging the “Product Features” information found on Amazon product pages (Najmi, 2015), or through utilizing a language model (Wu 2009).

Our goal was to design a framework that could be applied independent of data source where the only required input were the actual free text product reviews. Therefore, we opted for a more rule based system.

This requires an assumption that aspects are discussed in grammatically similar ways such that linguistic rules can be effectively applied, which research has shown to hold true (Liu, 2010). Liu’s research identifies that there are two key types of aspects: 1) explicit (or direct opinions) and 2) implicit (or indirect opinions). Mining implicit aspects is significantly harder than mining explicit aspects.

There is a significant body of work that utilizes this underlying grammatical and dependency structure to extract product. In Wu et al, (2009), their team generates dependency phrase trees and extracts noun phrases and verb phrases relevant to aspects. Additionally, multiple teams (Poria, 2014; Pekar, 2014; and Maharani, 2015) utilize a series of expert defined grammatical and linguistic rules to extract product aspects with very positive results. The effectiveness of dependency parsing is also demonstrated by its heavy use in opinion mining (Zhi et al, 2018; Caro and Grella, 2013)

Sentiment analysis

[[Cedric to fill]]

* [[Bing Liu’s seminal sentiment analysis paper]]
* [[ULMFiT paper by Howard and Ruder]]
* [[There are literally a million papers to choose from on sentiment analysis]]

Aspect Ranking

There are typically two approaches to ranking extracted aspects and their corresponding opinions: 1) some variant of a weighted average of opinions and frequencies for each aspect and 2) a probabilistic ranking algorithm. While we employ both approaches in our final evaluation, we focus primarily on the probabilistic ranking algorithm.

For a probabilistic ranking algorithm to work, we must assume the following: a reviewer, when deciding to leave a review, first knows what aspects they plan to comment on, second they leave their review discussing these aspects, and finally, the overall score attributed to the product is a weighted aggregation of the individual opinions on each aspect. The goal of a probabilistic ranking algorithm is to then uncover the weights used in the weighted aggregation. All other data is considered known due to inclusion in the product review (overall score) or the sentiment analysis (aspect specific opinions). Thus, in past research it is common to use a variant of the expectation maximization algorithm (Zha et al., 2014; Yu et al., 2011), where the weights are considered the latent variables.

**Data**

Our project leverages millions of unstructured Amazon product reviews developed by Julian McCauley and his team for [[insert the two citations]]. The Amazon Product reviews data consists of two primary datasets: 1) Product Reviews and 2) Metadata. Both of which are utilized.

After De-duplicating, the Product Reviews dataset includes 88 million observations. Within our analysis we utilize the following columns:

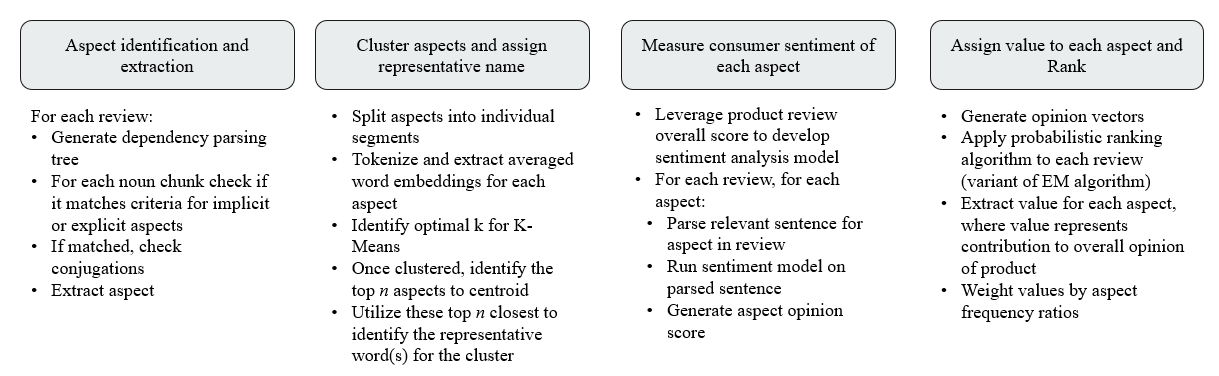
* ASIN: Product ID number
* Reviews: Unstructured review text
* Overall: number of stars (i.e., score) given to the product for the review

Additionally, the Metadata dataset consists of approximately 9 million products. From this we utilize the following columns:

* ASIN: Product ID number
* Categories: Complete list of categories associated with the product

Note, that due to time constraints, we have limited the number of reviews for each top-level category to 1 million. We incorporate [[33]] top-level categories in our analysis.

**Methodology**



*Figure 2:Detailed Analytical Pipeline*

Our approach consists of four discrete tasks:

1. Aspect identification and extraction
2. Cluster aspects and assign representative names
3. Measure consumer sentiment of each aspect
4. Assign value to each aspect and rank

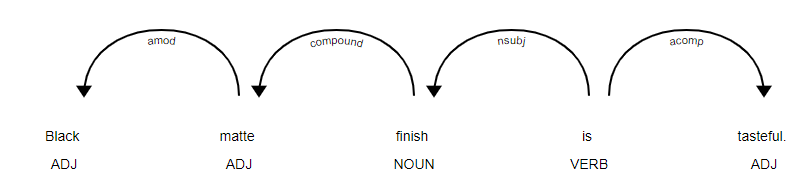
An overview for each discrete task is detailed in figure 2 above. The rest of our methodology section dives into each task in significantly more detail.

Aspect Identification and Extraction

As discussed in Liu, 2010, we assume that product aspects are discussed in linguistically and grammatically similar ways within product reviews. Liu writes that a direct opinion is represented as a quintuple vector (oj, fjk, ooijkl, hi, tl), where o*j* is an object, fjk is a feature of the object oj, ooijkl is the orientation or polarity of the opinion on feature fjk of object oj, hi is the opinion holder and tl is the time when the opinion is expressed by hi.

Our analysis leverages unstructured product reviews, therefore, we do not need to uncover all items within the quintuple vector. For example, the opinion holder (hi) is the username who wrote the product review and the time when the opinion is expressed (tl) is the timestamp associated with the review. Therefore, neither of these items need to be extracted from the unstructured text. Additionally, since product reviews are explicitly associated with a single object then oj is already known. However, we find that users still refer to the product by name or general reference. In conclusion, we find that the inclusion of object (oj) is optional in our direct opinion vector.

Following past research, we developed a rule based system to extract aspects. These linguistic rules developed according to Liu’s research on opinion mining are applied to the dependency parsing trees generated by SpaCy. In figure 3 below, we demonstrate the result of SpaCy’s dependency parsing algorithm on the sentence “Black matte finish is tasteful.”



*Figure 3: Dependency Parsing Tree on Sample Sentence*

For each review, we split the sentences using SpaCy’s sentence tokenizer, and generate a tree similar to figure 3. Once the tree is generated, we apply our set of linguistic rules[[1]](#footnote-1):

* Identify all noun chunks in the sentence
* Identify whether the noun chunk is connected to the rest of the sentence via a linking verb (“is”, “was”, “has”, “had”, etc.)
* Identify whether the connection is to another noun chunk or descriptor
* If the above rules match then we extract all noun chunks and / or descriptors
* Recursively apply rules to extract conjunctions

In the case of figure 3 above, it is apparent how these rules are applied:

* “Black matte finish” is our first noun chunk
* Which is connected to the rest of the sentence via a linking verb (“is”)
* The connection matches to a descriptor (“tasteful”)
* Therefore, we extract both “Black matte finish” and “tasteful” as our aspects
* There are no conjunctions, so we stop here.

Now, the reader may have noticed that we identified two sets of words that are quite different from one another (“black matte finish” and “tasteful”). This relates to differences between the two types of extracted aspects: 1) explicit and 2) implicit. In our analysis, we define *explicit aspects* as an aspect of the product that is directly mentioned. Additionally, we define *implicit aspects* as the implied product aspect from a descriptor. In the case of our sample sentence, “black matte finish” represents an explicit aspect that is directly discussing the finish applied to the product. Whereas, “tasteful” is defined as an implicit aspect where the aspect being implied is the design of the product.

While there are methodologies to map implicit aspects to their explicit form (i.e., map the word “cheap” to “price”) (Cruz-Garcia et al, 2014), we maintain implicit aspects in their current form. This allows the reader to make the decision of what the explicit aspect is at the time of evaluation.

Cluster Aspects and Assign Representative Names

After all aspects have been extracted successfully, our next goal is to group semantically similar aspects with one another such that we can assign a single name to word(s) that reference the same aspect. For example, we may have extracted two aspects “The battery life” and “the phone battery”. These two extracted aspects clearly refer to the same aspect “battery” – our goal with this section is to first identify that these two aspects are the same then assign each of them a common name for future analysis. We accomplish this through a two-step process:

* K-Means clustering
* Assign a common name using TF-IDF weights

First we generate the averaged word embeddings of each aspect then identify our clusters via K-Means for each top-level category. GloVe 100 dimensional embeddings and K-Means random initialization are used purely for computational speed. We limit the number of potential clusters to 100 or less for each top-level category. While, we understand that a higher number of clusters may lead to better clustering results, our work showed that the additional clusters created were small and would not be considered important aspects during our final analysis. Therefore, we felt that we were not losing much granularity by limiting to 100 clusters.

Second, we identified the most reasonable common names for each cluster. To accomplish this, we first subset each cluster to the closest 100 aspects to the corresponding cluster centroid. For this group of 100 aspects, we computed the TF-IDF scores and selected the top 5 words with the highest TF-IDF scores. These top 5 words were conjoined to form the single name representative of that particular semantic group.

Measure Consumer Sentiment of each Aspect

[[Cedric to fill]]

Assign Value to each Aspect and Rank

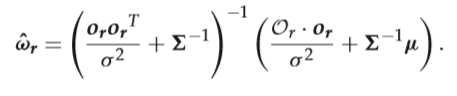
We take advantage of two separate approaches to measure the value or weight of each product’s aspects. The first is a regression analysis where the weights correspond to resulting coefficients [[Cedric to expand on regression summary]]. Additionally, we use the probabilistic ranking algorithm outlined in Zha et al. (2014). The results of both methodologies are compared and analyzed later in this report.

[[Cedric to add information on Regression]]

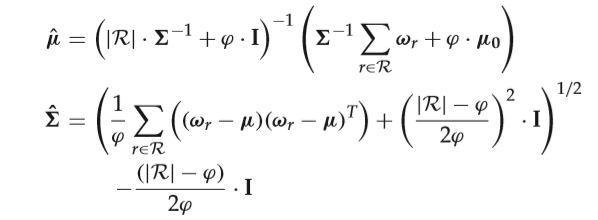
The probabilistic ranking algorithm is a variant on the expectation maximization algorithm designed specifically for the use of ranking aspects in free text product reviews. At a very basic level, the EM algorithm is used to iteratively infer latent variables through a series of updates when the probability distribution is known. It consists of two steps: 1) the expectation step where we estimate our latent variables and 2) the maximization step where we maximize the parameters of our distribution. In the case of our product aspect analysis, the latent variables that we are trying to infer are the product aspect weights. These weights, in other words, can be considered as the amount that each opinion for a specific aspect contributes to the overall opinion of each review.

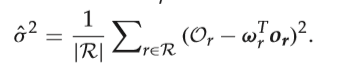
For a complete proof and derivation of the probabilistic ranking algorithm please refer to Zha et al. (2014). Below, we outline just the update equations for the expectation and maximization steps.

Expectation[[2]](#footnote-2):



Maximization[[3]](#footnote-3):



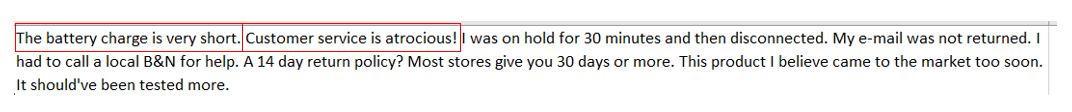


**Results**

In this section, we first outline the results of each our four discrete steps. Afterwards, we do a deep dive into the results of our analysis by displaying various results and any corresponding insights.

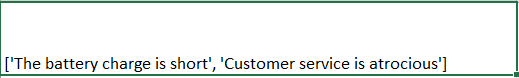
Aspect Identification and Extraction

Our team was able to successfully identify and extract aspects from all open text reviews. For details on number of aspects per category, please refer to Appendix [[A]] We first start with a raw review:



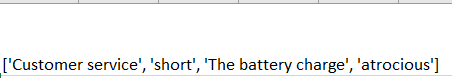
*Figure [[i]]: Raw Review with highlighted aspects*

Our aspect identification rules successfully identify the above aspects and extracts them into their own column:



*Figure [[i]]: Extracted Aspects*

At which point, we split these aspects into their individual aspects:

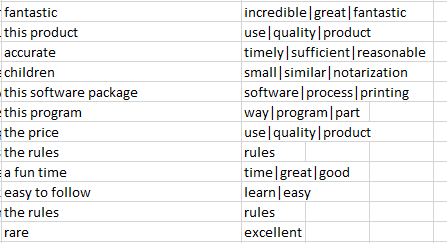


*Figure [[i]]: Extracted Aspects split into individual aspects*

Cluster Aspects and Assign Representative Names

Once aspects were successfully extracted, we clusters our aspects for each top-level category. For details related number of unique clusters per category, please refer to Appendix [[A]].

A sample of extracted aspects and their assigned names is displayed in the table below. Some aspects do not consist of 5 words, which is a result of subsetting the choice to the closest 100 aspects. Often times, we find that the closest 100 aspects are one or two aspects repeated over and over.



*Table i: Sample of Names assigned to Clusters [[to update with recent outputs and a shorter list]]*

Measure Consumer Sentiment of each Aspect Results

[[Cedric to fill]]

Assign Value to each Aspect and Rank Results

[[Put all visualization and inferences here]]

**System Overview**

[[Cedric to fill]]

**Future Work and Considerations**

We understand that while we believe that our results are significant, there is still room for improvements. We will use this section to highlight a few of the immediate improvements that we would like to make.

We would like to improve our initial aspect extraction algorithm. We understand that we simplified some of the linguistic rules surrounding our aspect extraction algorithm to increase interpretability. This likely has led to the inclusion of some noise (i.e., including words or phrases in our results that are not necessarily aspects). Additionally, we do not treat explicit and implicit aspects differently and treat them as a single entity throughout the entire analysis. Finally, when a descriptor is applied to an aspect (e.g., “short” in “The customer service is short”), often times it should not be considered as an implicit aspect. It is only when the descriptor is applied to the actual object (e.g., “large” in “the phone is large”) that we would want to consider it an aspect. A recent update to our underlying data includes a description of each product. We would wish to use this description field to identify when extracted aspects are discussing the actual object and only then would we like to extract descriptors.

Next, we would like to incorporate a variety of improvements to the clustering portion of the algorithm. First, we believe that leveraging a more sophisticated clustering algorithm (such as DBSCAN) would results in significantly improved results. DBSCAN is capable of acknowledging and ignoring outliers during the clustering process, of which we believe there are a lot. Finally, after clustering – we would like to have implemented an algorithm that identifies and remove intra-cluster outliers. Since we limit our analysis to at most 100 clusters, it is very likely that some semantic groups include aspects that are not particularly relevant to the assigned common name. Through removing outliers within each cluster, we can reduce the likelihood that we are over-weighting the importance of some aspects.

**Conclusion**

[[Short single paragraph concluding the results]]

1. This is an overly simplistic review of our applied linguistic rules. For a more detailed discussion, please refer to Poria, et all (2014) [↑](#footnote-ref-1)
2. *w*r is our weight vector for a specific Review (r). or is the opinion vector for each of our aspects for a specific Review. Or is the overall opinion of a specific Review. [↑](#footnote-ref-2)
3. µ is the mean of our gaussian (each aspect mean initialized as their corresponding frequency ratio). Σ is the covariance matrix. σ2 is the overall variance. ϕ is a constant. **I** is a identity matrix. |R| is the number of total reviews. [↑](#footnote-ref-3)