IDENTIFYING EXPERT REVIEWS IN THE CROWD: LINKING CURATED AND NOISY DOMAINS

A Thesis

by

ANIKET SANJIV BONDE

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	James Caverlee
Co-Chair of Committee,	Xia Hu
Committee Member,	Xiaoning Qian
Head of Department,	Dilma Da Silva

December 2018

Major Subject: Computer Engineering

ABSTRACT

Over the past decade, vast number of online consumer reviews have made a significant presence on the Internet. These reviews play a vital role in consumer awareness about the products and deeply impact the consumer's decision-making process. On one hand, websites like Amazon, Yelp provide huge collections of crowd-sourced reviews, which are written by consumers themselves having experience in using that product. Many researchers argue about the credibility and bias of these reviews. These factors, coupled with the sheer plethora of reviews for each product, it can become tiring to form a perspective about the product. On other hand, websites like Wirecutter, Thesweetsetup provide hand-made highly curated detailed guides on products across various categories. Although these reviews are unbiased expert opinions, they require vigorous reporting, interviewing, and testing by various journalists, scientists, and researchers. Thus making them hard to scale.

Our aim is to study the possible correlations between the crowd-sourced noisy domain reviews and the curated reviews. We take into account meta-features of reviews, context-based textual features of reviews and word-embedding based features of words from reviews. In addition to this, we identify "good reviews", defined as those noisy domain reviews that align with the curated ones, and use this to propose a general purpose, extremely streamlined recommender that can provide value to the general public without any personalized inputs. This research will contribute significantly towards identifying unbiased crowd-sourced reviews that align with curated reviews, across different categories of products, thereby linking the curated and noisy domains. Our research will also contribute significantly towards understanding the intricacies of good product reviews across different categories.

CONTRIBUTORS & FUNDING SOURCES

Contributors

The work was supported by a thesis committee consisting of Professor James Caverlee, Professor Xia Hu and Professor Xiaoning Qian.

Funding Sources

There are no outside funding contributions to acknowledge related to the research and compilation of this document. Although no explicit scholarship, my graduate study was partly sponsored by the funding from the grader work I did for three semesters in Computer Science and Engineering and Electrical and Computer Engineering departments.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

The e-commerce market has been booming over the last few years. This has led to a significant increase in time spent by consumer on e-commerce websites [4]. Websites like Amazon, Yelp, and TripAdvisor provide transparency to consumers by allowing users to give feedback in the form of reviews. Prospective consumers use these reviews to form a perspective about the product they want to buy. There is a significant increase in time spent by consumers on such crowd-sourced review websites as they prefer to research about the products they intend to buy [5]. These reviews add yet another level of transparency to the attributes of products. Additionally, sellers also get aided by the review feedback supplied by consumers on their products or their service itself.

1.1 Noisy Domain Reviews

Almost all crowd-sourced e-commerce platforms hugely rely on consumers who buy a product, to share their experiences with other consumers for products across categories. Websites like Amazon and Yelp have accumulated enormous amount of reviews owing to their popularity and product diversity [6]. These reviews range from highly informative ones to utterly uninformative ones. Amazon uses 'Helpful Votes' as the scoring method that provides information about how many people found that particular review helpful. New consumers often tend to refer reviews with most "Helpful Votes" to form a clear perspective about the product. However, in recent years, the issue of posting fake reviews has been elevated. Due to this inherent bias or spam, its imperative to extract trustworthy information and difficult to form a clear standpoint on products. Thus, various such biases in reviews [7] affect product sales and create an unfair marketplace for customers as well as sellers. This bias,

coupled with the enormity of number of reviews make this situation even worse.

1.2 Curated Domain Reviews

Although, this huge availability of reviews might seem as a boon, consumers often get overloaded with information making their buying decisions hard. Consumers then tend towards curated product review websites like Wirecutter, Thesweetsetup, and others. They provide hand-made highly detailed reviews on various products across categories. Wirecutter approximately reviews the top-100 products in each category. The reviews are made through vigorous reporting, interviewing, and testing by teams of journalists, scientists, and researchers. The review writers and editors are never made aware of which companies may have established affiliate relationships with Wirecutter's business team prior to making their picks. Hence, they claim their reviews to be unbiased. Each category has various articles which focus on one or more attributes pertaining to that category's products. They also recommend one or two best items looking at various attribute reviews and comparing them across all products in that category. Their reviews also enlist the possible competitor products with corresponding justification of why they were selected as competitors. Such websites provide concise information about product coupled with contrasting them with similar products, giving consumers an additional layer of transparency. Although these reviews are unbiased expert opinions, they require vigorous human interference, essentially making them hard to scale.

1.3 Goals

Our aim is to study the possible correlations between the crowd-sourced noisy domain reviews and the curated domain reviews. For noisy domain reviews, we considered the reviews on Amazon, as it is one of the biggest online e-commerce company. Figure 1.1 shows a typical amazon review. For curated domain, we used

review guides from Wirecutter. Figure 1.2 shows the Wirecutter recommended pick example with the justification review, for the same product. It is of interest to look at how the noisy crowd-sourced reviews are correlated with the curated ones for various products in a category, as well as how they vary across categories. Although there are various common aspects in reviews such as quality, cost and durability, that prevail across most categories, there are certain aspects to reviews that are unique to each category. Such similarities and differences learned across categories are used to identify and recommend "good products" such as those quoted "Top picks" and "Our picks" by Wirecutter.

Figure 1.1: Amazon review example [1]

We aim to explore features which will point out key similarities and differences between curated and noisy domains, thereby providing valuable insights in moving from noisy domains towards curated domains. Additionally, these features would help identifying informative and unbiased reviews from the noisy domain reviews. The features can also be exploited to form various models which could be leveraged to extract curated-domain-like information from noisy domains without actual human interference. For instance, we will try to predict how likely is a particular prod-

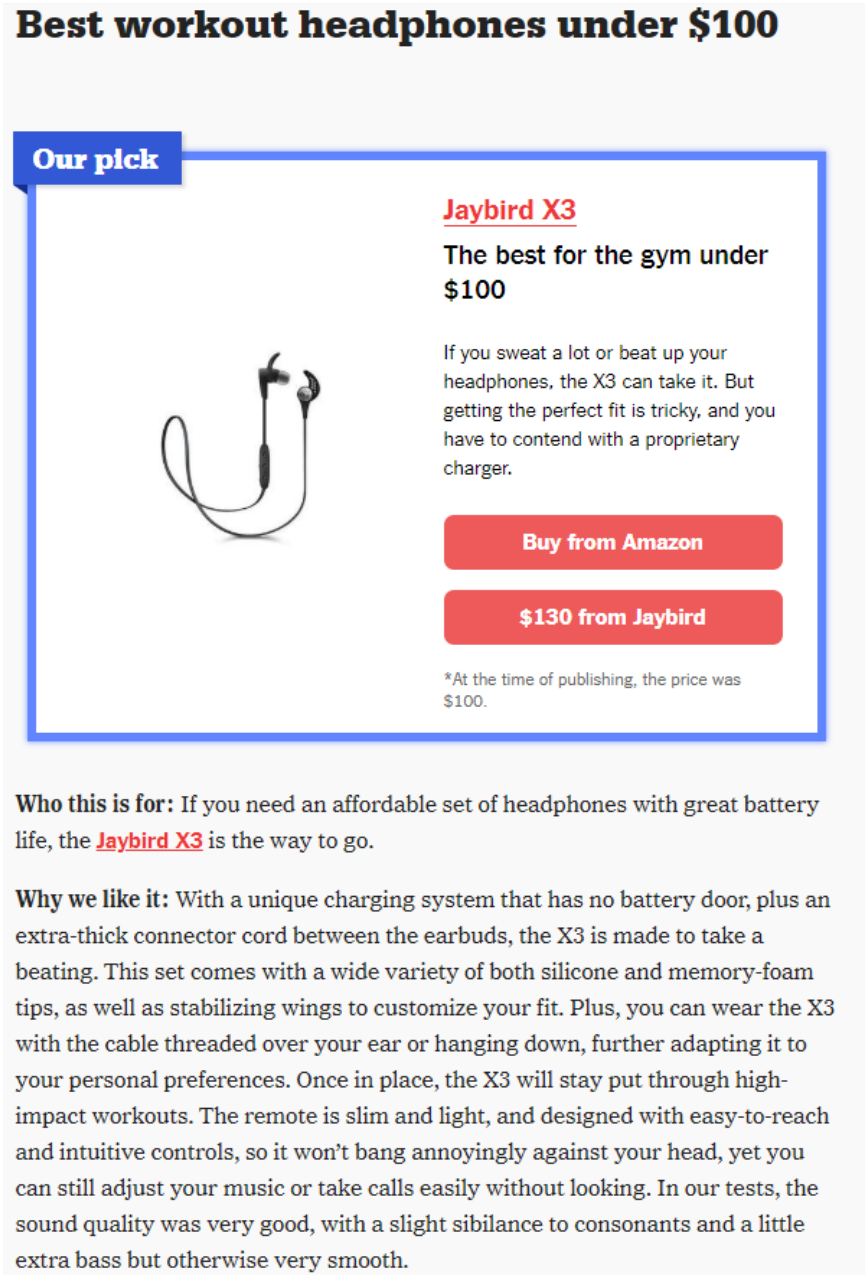


Figure 1.2: Wirecutter recommended pick example for 'Best workout headphones under $100' article in 'headphones' category [2]

uct from Amazon labelled as one of the "Top picks" by Wirecutter in a particular category, solely leveraging features from noisy domain.

More specifically, we formalize our goals as follows:

Wirecutter recommends some of the reviewed products as the best ones. We address them as *wirecutter_selected* and the rest ones as *wirecutter_unselected*. Now, we aim to predict the likelihood of a particular product on Amazon to fall into those classes, solely leveraging review aspects from Amazon. Thus we propose a general purpose, extremely streamlined recommender that provides value to the general public without any personalized inputs. Additionally, accurate predictions would ensure that some aspects of noisy domain reviews (from Amazon) are analogous to curated reviews (from Wirecutter). Utilizing such model, we intend to explore what aspects of the *wirecutter_selected* product reviews differ from the *wirecutter_unselected* ones in the noisy domain. Moreover, we aim to identify reviews that play a major role in determining product as *wirecutter_selected* or *wirecutter_unselected*, and to inspect such reviews to see if they have similar characteristics as the Wirecutter review. Reading only such reviews would help consumer to make better perspective about a product rather than by reading top most helpful reviews, as review helpfulness is merely a measure of how helpful a particular review is and not how "good" the reviewed product is. We will further examine whether using only expert reviews increases accuracy for the prediction task stated earlier.

This thesis aims to address the following research questions:

- Are there any key factors (statistical or latent) that are common to curated and noisy domains? If yes, can they be leveraged to move from noisy domains to curated ones without human interaction, essentially making them easy to scale?

- Determine if these key factors are consistent across all product categories. If not, what's the difference?

- This research also sheds some light on identifying informative and unbiased reviews. Determine what aspects makes a review informative and unbiased? Are these aspects consistent across all categories?

1.4 Challenges

In this section, we describe the key challenges faced during this research. We also elaborate on how we addressed these challenges.

1.4.1 Minimal Prior Work

Although there has been a lot of research in the linking of several domains. However, the direction of research that we are heading has still been relatively untouched. Hence, we found it arduous to find any related work to this research. The work in [8] has motivated us in terms of idea. Dealing with crowd reviews is more about storing and processing them as raw text documents which stands as the underlying definition of Natural Language Processing. Natural Language Processing is still a hot topic and its future will be redefined as it faces new technological challenges and a push from the market to create more useful and user-friendly systems [9]. This dependence on the technological and market obstacle on the processing side has restricted our options. Due to minimal prior research in this field, the task is challenging.

1.4.2 Limited Overlap Between Multiple Domains

In order for us to perform our analysis, we needed a noisy domain (user-contributed crowd reviews) and a hand-made curated domain (expert opinion reviews) with constraint – huge overlap in terms of the products they review upon. However, different websites review different products, with limited categories having dense overlap. Therefore, it became unsuitable to conduct our experiments on sparsely overlapping categories. Additionally, we also wanted categories in the crowd domain to have avail-

ability of large number of reviews, so we could obtain a good set of unbiased reviews by applying various similarity measures between both domains. We also wanted to maintain diversity in our dataset in terms of number of reviews per category. Does more crowd-sourced reviews necessarily convey more in-depth unbiased information about product? Does the factor "number of reviews" positively correlate with the quality of product? These questions can only be answered if we have categories with number of reviews ranging from low to high. It would also help us investigate the variation in the influence of factors determining how good the product is.

2. RELATED WORK

This section is focused on understanding related research literature on linking various online product review domains. However, only few of these multi-domain studies involve linking curated and noisy domains, and fewer on how identifying curated-like product recommendations leveraging only noisy domain reviews. We have already discussed the scarcity of prior work as one of the challenges of our research in Section 1.4.1. We will also describe how our work differs in context to the existing approaches.

2.1 Multi-Domain Studies

A few recent studies [8] [10] [11] [12] are concerned with transferring information between various domains. This is related to our goal. [8] study the problem of inferring the more calibrated "expert ratings" from the user-contributed ratings. To achieve this, they employ latent factor models and provide a probabilistic treatment of the ordinal ratings. They predict expert ratings accurately from ad-hoc user generated noisy ratings by employing joint optimization. Furthermore, by the resulting model they conclude that users become more discerning as they submit more ratings. [10] [11] [12] use collaborative filtering techniques on review ratings for transferring information between various domains.

[13] exploit the variety of huge crowdsourced or user-generated data and find trustworthy insights without domain expert, using a multiple source mining approach. They use the knowledge crowdsourced and transferred from external domains. They study the strengths and weaknesses of various knowledge transfer strategies and propose Consensus Ranking Dual Transfer (CRDT) to handle the multiple data source challenges like 1) inherent heterogeneity, 2) partial overlapping

nature and 3) biased by themselves, by identifying "anchor reviewers" as a bridge for robust "dual transfer", and removing bias in individual sources via consensus ranking aggregation.

2.2 Expert Unbiased Reviews Identification

Later phase of our research deals with identifying expert reviews from crowd reviews by leveraging curated review. So, we reviewed related literature to study the previous work in this domain.

[14] proposed a metric that signifies the quality of a review by accounting for customers personal experiences – by measuring their 'mentions about experiences'. They have used a rule-based classifier to perform sequential checking of syntax, cue, tenses, time expressions. The selected reviews by their metric are as helpful as the reviews selected by the number of helpful votes, without biases.

Various other studies like [15] [16] [17] [18] use the helpfulness measure as a means to quantify how informative or descriptive the particular review is. [15] extracts informative user reviews by filtering noisy and irrelevant ones then grouping the informative reviews automatically using topic modeling. They further prioritizing the informative reviews by an effective review ranking scheme, and finally presenting the groups of most informative reviews via an intuitive visualization approach. They calculate scores for each of the reviews by topic modelling. [16] studies the impact of the various features, that is, basic, stylistic, and semantic characteristics of online user reviews on the informativeness of the review. [17] use Helpfulness in various online communities as a measure of message quality. They perform factor analysis to show five underlying quality dimensions that are representative of informativeness of the review: reviewers reputations in the community, the topical relevancy of the reviews, the ease of understanding them, their believability and objectivity.

[18] built a theoretical model based on three elements: conformity, understandability and expressiveness and investigate the directional relationship between the qualitative characteristics of the review text, review helpfulness and the impact of review helpfulness on the review score. Furthermore, they also examine whether this relation holds for extreme and moderate review scores.

[19] identifies trustworthy reviewers and states that trustworthy reviewers give informative reviews. They use Amazon rank, the reviewers total number of posted reviews, and three user-derived features: the number of reviewer's Amazon badges, the amount of personal information revealed on reviewer's profile, and the reviewer's average posting frequency to build a probabilistic model that predicts the trustworthiness of a reviewer on Amazon. [20] identifies informative unbiased reviews from product designers' viewpoint. They use regression techniques on linguistic, product information and review sentence sentiment orientation to determine the informativeness of review.

2.3 Overview

Additionally, most of the literature that deals with transferring information between domains uses ratings. We aim to explore meta-features of reviews, context-based textual features of reviews and word-embedding based features of words from review along with the review ratings. Hence, these features will point out key similarities and differences between curated and noisy domains, thereby providing valuable insight in moving from noisy to curated domains. Resulting models can be leveraged to extract curated-domain-like information from noisy domains without actual human interference. This coupled with minimal prior work led us to believe that the problem statement we are dealing with is challenging indeed. This thought has heavily inspired us.

3. DATA COLLECTION

In this chapter, we describe the various sources for data collection, how we obtained them and the challenges faced during the whole process. We also elaborate on how we addressed these challenges while ensuring that our approach remained most suitable to address our goals. This chapter also sheds some light on the reasons behind choosing Amazon and Wirecutter as the sources of our dataset.

3.1 Need for Scraping: Absence of Dataset with Overlapping Expert & Crowd Domains

There are a lot of e-commerce websites that provide crowd-sourced user generated reviews. Along with this, there also exist a fair amount of websites that provide curated reviews on various products. Our main concern is to have huge overlap in terms of the products they review upon. However, different websites review different products, with limited categories having dense overlap. Therefore, it became unsuitable to conduct our experiments on sparsely overlapping categorical dataset. In addition to this, the field of linking noisy and curated domains is relatively untouched. Therefore, we found it arduous to locate any datasets for the research to be conducted upon. Due to this reason, we decided to collect data on our own.

3.2 Data Sources Selection for Expert & Crowd Domains

As curated review websites are less in number as compared to crowd-sourced ones, we firstly needed to decide on the data source for curated domain. There are various websites like TheSweethome, Wirecutter, Best best list, Buzzfeed reviews, Thesweetsetup and many more, that provide expert opinions on products. We decided to initially restrict our data set to wirecutter.com for the following reasons:

- We wanted to initiate our research with a single and reliable source
- Wirecutter is the largest online source of curated reviews. It links heavily to outside sources like expert reviews, consumer reports and independent product reviewers, and includes information from those experts in its recommendation essays. It was launched in 2011. In the five years from 2011 to 2016, the company generated $150 million in revenue from affiliate programs with its merchant partners

Wirecutter hyperlinks all the products either to the product's website or to famous e-commerce websites like Walmart, Amazon, and many more. Most of their links lead to Amazon. Additionally, Amazon has one of the largest database of consumer opinions in the form of reviews across numerous categories as compared to other e-commerce platforms. Its also categorically diverse and has metadata information (timestamp, productID, helpful votes) on each review. All this, coupled with the immense variation in review types, lengths, and reviewers led us to believe that Amazon would be ideal noisy data source for our research.

3.3 Scraping

The need for web scraping is declining with the vast increase in web services. But, there are situations when web scraping is useful:

- When there is restriction on rate and volume of requests or API yields unsuitable formats and types of data [21].
- There is restriction on access of desired API services
- Independent web services with little scope for interoperability

- Operational cost of understanding API usage when such an investment is not justified, for example, during prototyping or source evaluation [22]

Web Scraping is the method of gathering data from the Internet through any means other than a program interacting with an API or a human using a Web browser. This can be achieved generally by writing an automated program simulating human exploration of the Web that queries a web server, requests data and parses that data to extract required information [21] [23] [24]. Following are the forms of scraping:

- Web Scraping - Unstructured data from the web is extracted and processed into structured data to be stored in a database.
- Screen Scraping - The output of a program is extracted as result for the end user instead for another program (usually for legacy applications with obsolete Input/Output Device or interface)

There are many ways to scrape the Web. This includes human-copy paste (feasible for small-scale projects), text grepping using regular expressions, HTTP programming, DOM parsing, HTML parsers, and making scraper sites (Websites created from scraping contents from other websites) [24]. [25] gave the perspective of HTML pages as containing two tokens - HTML Tag tokens and text tokens and represented HTML pages using a sequence of bits (0 - text, 1 - HTML tag). However, this approach was applicable to single body HTML documents only and would not be a viable option for modern day multi-body HTML pages as it will take polynomial time for execution with a degree equal to number of bodies in the document. [26] used Document Object Model (DOM) tree for content extraction model by removing all the links from the page. But, this approach too is not usable for search engine

websites like Google and Bing and multi-page websites. As shown in Figure 3.1 and Figure 3.2, these two approaches cannot be used for scraping wirecutter.com.

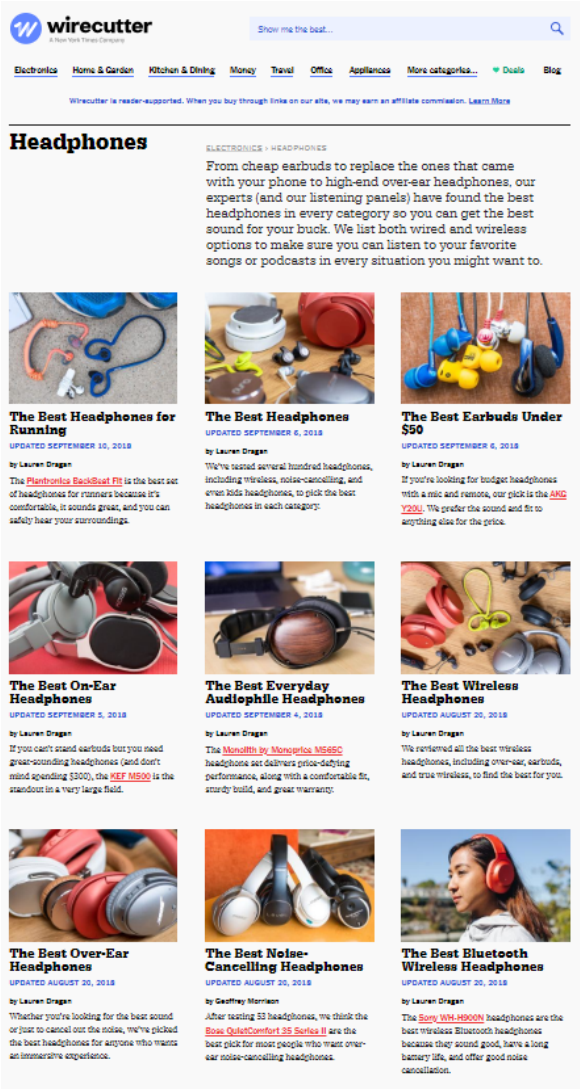


Figure 3.1: Overview of wirecutter.com website: Homepage of wirecutter.com shows navigation links to multiple web-pages [2]

Wirecutter organizes its content by various categories. Each category has its catalog of posts, where each post comprises of detailed essay reviewing products by

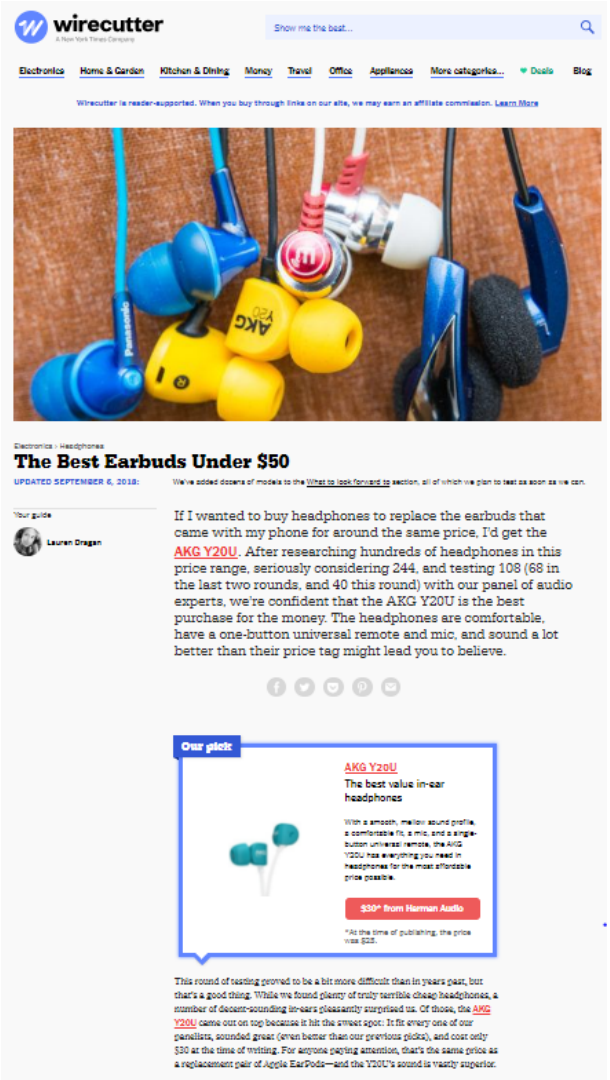


Figure 3.2: Overview of wirecutter.com website: Wirecutter expert review on 'The best earbuds under $50' with their top picks [2]

similar features, price range, etc.

Figure 3.3 shows a more zoomed view of three posts from 'headphones' catalog. Each post is an hyperlink to a page which reviews various products and marks two or three products as 'Wirecutter top picks'

As Wirecutter does not have any underlying API for fetching the data off the

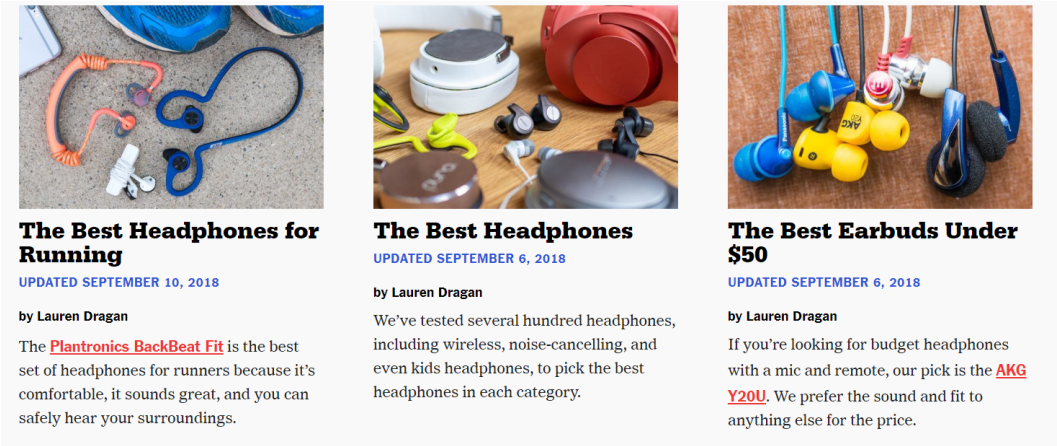


Figure 3.3: Wirecutter posts [2]

website, we extracted data through scraping. We employed custom crawler based on HTML parsing to scrape review and other data from Wirecutter.

Amazon also does not provide any API for fetching the reviews. Hence, we deployed two types of crawlers:

- As the number of pages to be scraped from wirecutter.com were less, Scrapy is a good tool written in python. If the number of pages to be scraped is too large, Scrapy would end up with *Maximum retries exceeded with url.*

  Scrapy: Its a Fast and Powerful Scraping and Web Crawling Framework on python. We used this framework to scrape wirecutter.com. Both, the main pages and sub-pages were scraped with different scrapy programs.

- Although there are various huge Amazon reviews databases openly available, we decided to write our own crawler because the databases were outdated and the Amazon products did not overlap with the new Wirecutter ones. Hence, Amazon.com is the source of our crowd-sourced noisy reviews. As there are numerous products on this website under each category, it becomes hard to

scrape all reviews for all products. Hence, for initial phase of our research, we decided to scrape only two types of products:

- All the recommended top picks by wirecutter.com, that hyperlink Amazon.com. We dub these products as *wirecutter_selected*.
- All products taken into consideration while choosing those top picks on wirecutter. These products do not include the recommended picks stated in the former point. We dub these products as *wirecutter_unselected*.

As there were $\sim 60,000$ Amazon.com pages to be crawled, Scrapy gave the error *Maximum retries exceeded with url*. Hence, we devised custom crawler based on HTML parsing that used tor for making requests to urls. We toggled ip and tor nodes after two or three requests to avoid getting blocked by Amazon.com. But, despite such configuration, Amazon.com hindered the scraping after $\sim$150 requests with a captcha. We then devised a captcha solver using Python Tesseract Optical Character Recognition library. We scraped 525582 reviews for 1040 products.

Wirecutter, being hard to scale, only have detailed guides for a handful of product categories. Hence, we restricted our research to include models for those categories only. More specifically, we restrict our models to data from Amazon and Wirecutter for four product categories: headphones, cameras, laptops, projectors.

3.4 Dataset Details

Each review is represented by single json dictionary structure as shown in Figure 3.4.

The JSON key definitions are provided below:

- **_id**: Database storage ID for this review

```
{
  "_id": "5a6fe4e1bf5a84161caebff6",
  "title": "Great Sound but (sound sync issues)",
  "rating": "3",
  "body": "I'm using this with my iPhone 7 plus and noticing a delay in sound (sound sync issue).",
  "product_id": "B01HRYAP1K",
  "date": "October 16, 2016",
  "helpful_statement": "62 people found this helpful"
}
```


Figure 3.4: Sample Amazon Review JSON data.

- **product_id** : Amazon encrypted product ID
- **title** : The summary title of the review as written by the reviewer
- **body** : The review in text format
- **rating** : The product rating as given by the reviewer. It lies between the range [0, 5]
- **date** : The time the review was posted (raw format)
- **helpful_statement** : Number of users who voted the review as 'helpful'

Figure 3.5 shows the metadata information for a particular product id. There is exactly one metadata JSON object for each product.


```
{
  "_id": "5a6fc68317ada517820232a5",
  "product_id": "B01HRYAP1K",
  "source": "wirecutter_selected",
  "no_of_reviews_collected": 3007
}
```


Figure 3.5: Metadata JSON for a Product ID.

The JSON key definitions are provided below:

- **product_id** : Amazon encrypted product ID
- **source** : This indicates if the product was selected by Wirecutter as one of its 'top picks'. Possible values are ['*wirecutter_selected*', '*wirecutter_unselected*']
- **no_of_reviews_collected** : The total number of reviews scraped from Amazon for this product. We collected all reviews for every product we considered.

Table 3.1 shows the information about the data we collected across all categories.

Categories	Amazon (*wirecutter_unselected*)	Amazon (*wirecutter_selected*)	Total products (reviews)
Headphones	533 products (232,486 reviews)	69 products (103,157 reviews)	602 products (335,643 reviews)
Cameras	115 products (57,049 reviews)	166 products (61,742 reviews)	281 products (118,791 reviews)
Laptops	30 products (22,242 reviews)	82 products (38,609 reviews)	112 products (60,851 reviews)
Projectors	11 products (2,282 reviews)	34 products (7,475 reviews)	45 products (10,297 reviews)

Table 3.1: Dataset Specifications

4. METHODOLOGY AND MODELS

We initiate our research towards answering this question:

Are there any linkages between noisy and curated domains? If yes, what are the key factors (statistical or latent) common between both domains? To answer this question we explore various crowd domain review attributes and how they correlate with those of curated domain.

We formulate our goal as follows:

Wirecutter recommends some of the reviewed products as the best ones. We address them as *wirecutter_selected* and the rest ones as *wirecutter_unselected*. Now, we aim to predict the likelihood of a particular product on Amazon to fall into those classes, solely leveraging review aspects from Amazon. Thus we propose a general purpose, extremely streamlined recommender that provides value to the general public without any personalized inputs. Additionally, accurate predictions would ensure that some aspects of noisy domain reviews (from Amazon) are analogous to curated reviews (from Wirecutter). Utilizing such model, we intend to explore what aspects of the *wirecutter_selected* product reviews differ from the *wirecutter_unselected* ones in the noisy domain.

We now articulate various features to capture the correlation.

4.1 Feature Engineering

After the data collection as described in section 3.3, we focus on feature engineering aspect. We aim to explore three specific domains:

4.1.1 Rating and Text Statistics based Features

These are the meta-features from the star-ratings such as the average of ratings, median, absolute deviation from mean and variance, etc [27]. The features also entail average of helpful votes, review body length and review title length for all reviews of a product. We also analyze features in the review title.

4.1.2 Context based Textual Features

These features are focused on text in reviews. They entail the tf-idf vectors calculated by combining all reviews for each product.

4.1.2.1 Tf-idf Overview

Term frequency inverse document frequency is bag-of-words model used to represent how important a word is to a document in a collection or corpus. More common words across review documents convey less information as compared to rare specific words that may be present in very few review documents. The tf-idf score is product of two factors: the term frequency (tf) and the inverse document frequency (idf). Term frequency indicates how frequent a particular term is in a given review document. We use relative frequency of a term instead of the raw count. This is because relevance of a word does not increase proportionally with raw word frequency. Inverse document frequency indicates the degree of information the word provides, that is, whether the term is common or rare across all documents. Inverse document frequency is calculated by first computing the document frequency of a term as the number of review documents that contain that term, then logarithm of the ratio of total review documents in corpus to the document frequency of that term defines the inverse document frequency score. Later, product of these two scores is termed as tf-idf score.

It is important to note that the relative ordering of terms in the review document is not considered important while calculating tf-idf, only the presence of a particular term is noted. Hence, it won't take into account the semantic or syntactic relationships of words in reviews. Word sequences with complementary meanings can have similarity for tf-idf vectors, if the terms used are same. Additionally, the embedding vectors are highly sparse since vocabulary is much larger than the set of terms used in any review document. Such sparsity often generates dissimilarities between two reviews that may be similar in context, but different in usage of terms. Although minute disadvantages, it is widely used numerical statistic in the field of information retrieval and the basis of modern search engine algorithms.

4.1.3 Word-Embeddings based Features

The textual features are just about the level of common occurrence in the textual units. We are interested in identifying latent features from review context. Hence, we plan to use techniques such as topic modeling and 'word2vec' embeddings to fuse semantic and syntactic relationships into the model.

4.1.3.1 Word2Vec Overview

Word embeddings techniques are an excellent approach to language modelling, which generate dense word vectors, as compared to sparse tf-idf vectors. [28] proposes 'word2vec' which uses neural networks to get a vector representation of words in context. These representative word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space. It explicitly relies on the distributional hypothesis of semantics by attempting to predict the surrounding context of a word, either as a set of neighbouring words (the skip-gram model) or as an average of its environment (continuous bag of words). Figure 4.1 shows the widely used implementation

architectures for creating word embeddings.

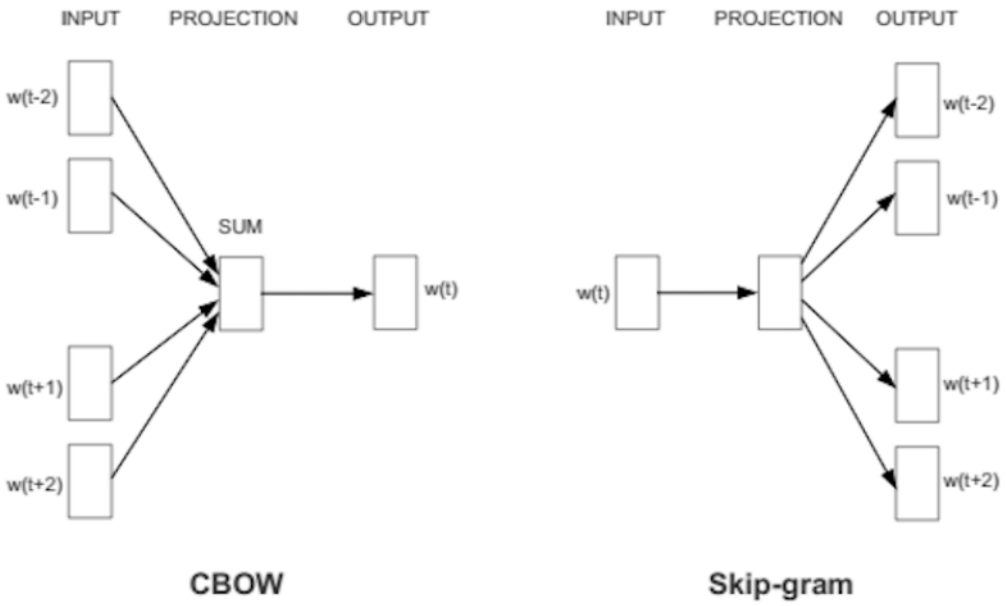


Figure 4.1: Word2Vec architectures to create word embeddings [3]

In the continuous bag-of-words architecture, the model predicts the current word from a window of surrounding context words. The order of context words does not influence prediction (bag-of-words assumption). In the continuous skip-gram architecture, the model uses the current word to predict the surrounding window of context words. The skip-gram architecture weighs nearby context words more heavily than more distant context words. [28] states that skip-gram represents well even rare words or phrases and continuous bag of words model is several times faster to train than the skip-gram with slightly better accuracy for the frequent words.

In our approach, we chose to proceed with skip-gram as it has been proven to be more accurate than other models, such as continuous bag of words, due to the more generalizable contexts generated. This model yeilds latent dense embeddings

for each word in embedding space. Using 200 latent dimensions for the skip-gram model addressed the trade-off between less training time and getting better word embedding representations.

4.2 Setup

Figure 4.2 shows generalized view of our approach to model building. We aim to predict the likelihood of a particular product on Amazon to fall into one of our defined classes, solely leveraging review aspects from Amazon.

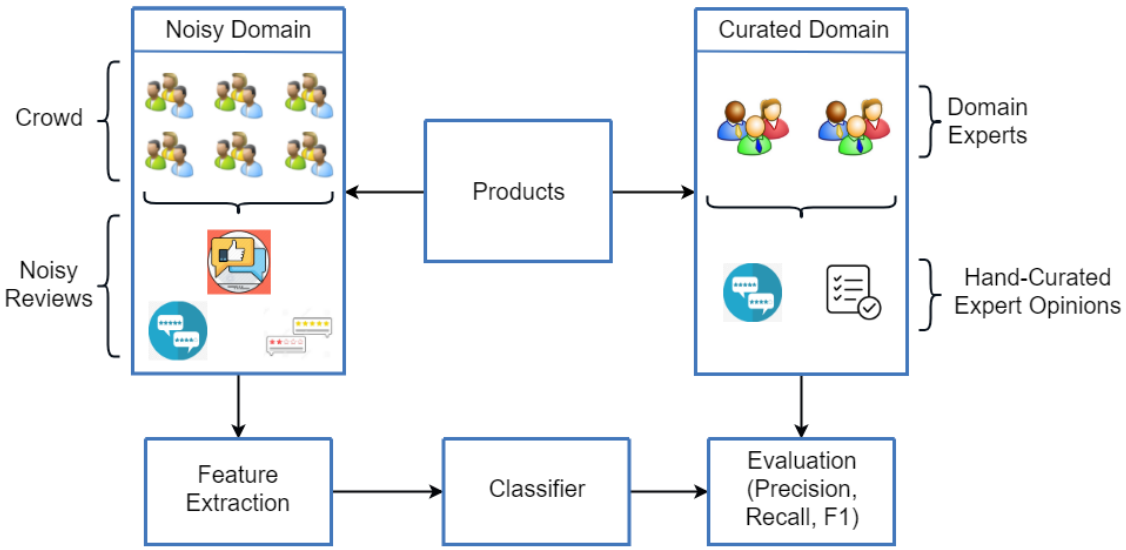
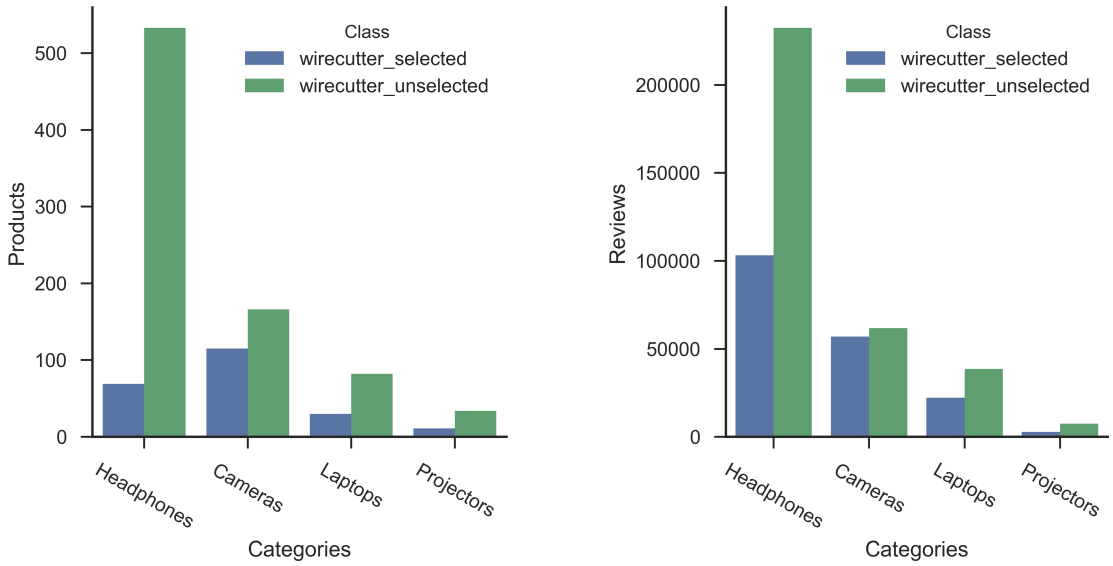


Figure 4.2: General Model Building Setup.



(a) Products

(b) Reviews

Figure 4.3: Class Distributions

We divided our research in two phases:

In first phase, we plan to use features (as describes in section 4.1) of all reviews in the model. The features are calculated on per product basis, to determine if the product would be recommended by Wirecutter as "our pick", solely based on noisy domain features, thereby linking the curated and noisy domains.

In second phase, we identify the reviews that align with the Wirecutter detailed guides. We dub these reviews as "informative and un-biased reviews". Now, these reviews will be used for further feature extraction. Again, the features are calculated on per product basis, to determine if the product would be recommended by Wirecutter as "our pick", solely based on noisy domain features, thereby linking the curated and noisy domains.

Then we plan to deploy various traditional machine learning models such as random forest, SVM, and others.

For the first phase, following is the process we followed:

- All the products are from Amazon.com in our dataset. The recommended top picks by wirecutter.com are called *wirecutter_selected* to indicated that these products are the absolute bests in the category.
- All the remaining products are called *wirecutter_unselected*. These are the products taken into consideration while choosing those top picks on Wirecutter, not including the top picks.
- Features are calculated on per product basis, meaning that the feature vector of a product represents aggregated features from all its reviews.
- We now compare the calculated features of *wirecutter_selected* class versus *wirecutter_unselected* ones to see any key similarities. So, feed the product feature vectors to a Random Forest Classifier to classify that product as *wirecutter_selected* or *wirecutter_unselected*.
- As the amount of data available is limited, we have used a 5-fold cross-validation scheme. It provides less sensitivity to the partitioning of the data and lower variance as compared to a single hold-out set estimator.
- The class distributions shown in Figure 4.3 the classes are highly imbalanced, hence accuracy is not the right metric for evaluation of our models. Therefore, Precision, Recall and F1 Scores are used for the evaluation. Figure 4.4 shows the general setup for baseline models on a category.

4.3 Baseline Models

What qualities define a good product? (Where we define a *good product* to be among the ones recommended by Wirecutter) To answer this question, we deployed

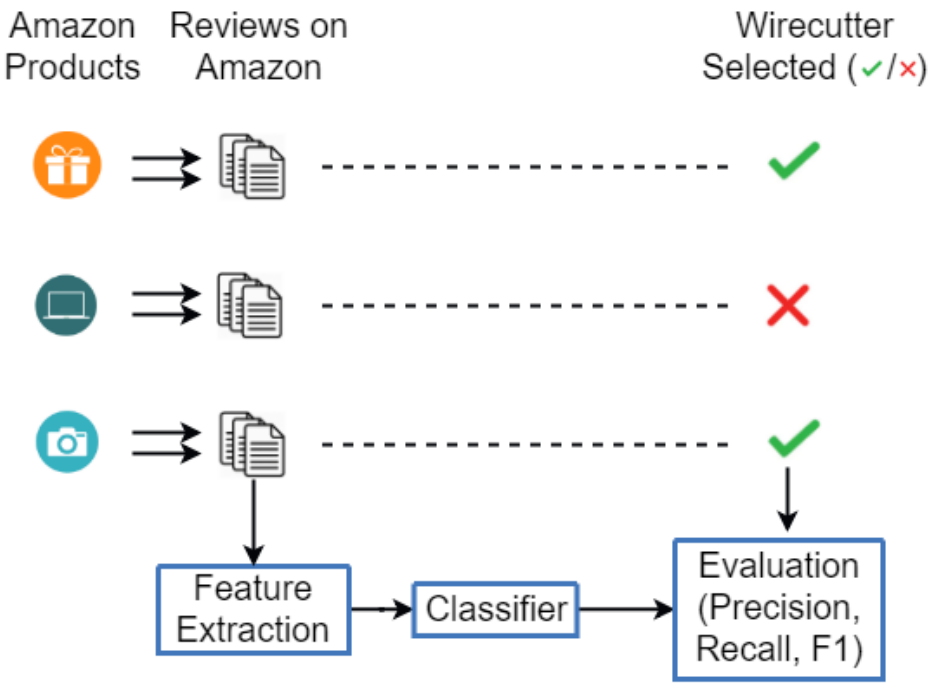


Figure 4.4: General Baseline Model Building Setup.

some basic models to capture the key differences between *wirecutter_selected* product reviews and *wirecutter_unselected* product reviews.

4.3.1 Meta-Features Model (MF)

For our first baseline model, we use rating and text statistics based features. These are the features from the star-ratings such as the average of ratings, median, absolute deviation from mean and variance,etc [27]. The features also entail average of helpful votes, review body length, review title length, review post latency (time elapsed between product launch and posting of a particular review) for all reviews of a product.

Table 4.1 shows the results of the above described setup. From here on-wards,

all the evaluation metrics would be calculated considering *wirecutter_selected* class to be the positive class.

Categories	Accuracy (%)	Precision	Recall	F1 Score
Headphones	84	0.40	0.13	**0.20**
Cameras	63	0.47	0.43	**0.45**
Laptops	60	0.30	0.28	**0.29**
Projectors	61	0.38	0.25	**0.30**

Table 4.1: Preliminary Classification Results using Random Forest Classifier with Meta-Features of Reviews and 5-Fold Cross-Validation Scheme

Despite good accuracies (due to imbalanced classes), the F1 score seems to be low. This might be due to the fact that the model lacks context-based features derived from the text of reviews. This led us to further explore the idea of leveraging text based features.

4.3.2 Context based Features Model (CBF)

We explore topic modelling as a means to generative construct topic distributions for product reviews across various categories. We decided to look if there are any common topics in both classes. We employ a popular topic modelling approach known as Latent Dirichlet Allocation(LDA) [29]. Results for two of the four categories are shown in Figure 4.5. The other two categories showed similar results. The red links signify similar clusters after Latent Dirichlet Allocation is carried out on *wirecutter_selected* and *wirecutter_unselected* product reviews individually.

The minor similarities signify that there are some features common between both domains indeed. This led us to further explore the idea of leveraging text based features. Apart from this, no other significant insights could be deducted.

Laptops (LDA):

Wirecutter Selected

Topic #	Top 10 representing words
1	book, smell, mac book, zipper, mac, book pro, mac book pro, excellent quality, chrome book, perfect macbook
2	battery, battery life, hours, wifi, power, windows, performance, ram, ssd, settings
3	service, customer, anker, customer service, send, contacted, replacement , update, charge, defective
4	hdmi, windows, ssd, usb, games, gaming, ram, adapter, samsung, fast

Wirecutter un-Selected

Topic #	Top 10 representing words
1	gaming, games, cpu, gaming laptop, ram, play, performance, game, processor, settings
2	return, cooler master, unable, works works, warranty, stars worked, wrist guard, desk large, wrest, cushioned
3	cooler master, notepal, blades, master notepal, cooler master notepal, dell, beads, xps, fan blades, dell xps
4	lap desk, desk, lap, rubber, height, book, wire, laptop stand, mstand, desk

Cameras (LDA):

Wirecutter Selected

Topic #	Top 10 representing words
1	camera, great, quality, good, video, use, lens, works, stars, just
2	lens, stars good, wide, sharp, focal, canon, fuji, focus, nikon, aperture
3	mac, support, software, linux, facetime , work mac, tech, gaming, mac mini, service
4	cleaning, lint, wipes, clean, free, lint free, solution, pad, dust, stream
5	mic, conference, stars amazing, built mic, external mic, microphones, hdmi, mics, camera mic, sound good
6	cam, software, audio, usb, plug, recording, drone, logitech, windows 10 , drivers

Wirecutter un-Selected

Topic #	Top 10 representing words
1	usb, reader, card, micro, card reader, sd card, sd cards, micro sd, slot, macbook
2	camera, great, use, good, quality, like, just, pictures, lens, video
3	takes great, nex, takes great pictures, g7x, peaking, powershot, focus peaking, takes great photos, sony nex, camera takes great
4	lap desk, desk, lap, rubber, height, bo ok, wire, laptop stand, mstand, desks

Figure 4.5: Latent Dirichlet allocation on two categories: Laptops and Cameras.

To leverage text based features, we calculated the Term Frequency Inverse Document Frequency (tf-idf) vector corresponding to each product. We combined all reviews for each product, calculated tf-idf vectors and used these vectors as the features for our model. The same Random Forest Classifier with 5-Fold Cross-Validation scheme is used. The results yielded are summarized in table 4.2

Categories	Precision	Recall	F1 Score
Headphones	0.33	0.21	**0.26**
Cameras	0.42	0.39	**0.40**
Laptops	0.44	0.33	**0.38**
Projectors	0.29	0.27	**0.28**

Table 4.2: Preliminary Classification Results using Random Forest Classifier with tf-idf features of Reviews and 5-Fold Cross-Validation Scheme

As compared to meta-features model, almost similar results were obtained in the tf-idf model as well. This model does not take the semantic or syntactic relationships of reviews, it's just about the level of common occurrence in the textual units to be learnt from. Hence, the next step was to exploit semantic and syntactic relationships of the words in reviews.

4.3.3 Word2Vec Embeddings Model (W2V)

To better understand the correlation of semantic or syntactic relationships of reviews between noisy and curated domains, we used word2vec [28] embeddings for our next model. Similar to our last model, we aggregated all reviews for each product and simply averaged [30], [31] the word2vec embeddings of all words to form a feature vector for that product. The same Random Forest Classifier with 5-Fold Cross-Validation scheme is used. The results yielded are summarized in table 4.3

Categories	Precision	Recall	F1 Score
Headphones	0.45	0.21	**0.29**
Cameras	0.40	0.33	**0.36**
Laptops	0.36	0.19	**0.25**
Projectors	0.29	0.23	**0.26**

Table 4.3: Preliminary Classification Results using Random Forest Classifier with Averaged Word2Vec Embeddings of Review words and 5-Fold Cross-Validation Scheme

Looking at the table 4.3, its clear that results did not budge much. These preliminary results led us to believe that the problem statement we are dealing with is a hard and non-trivial problem indeed. This thought process got us even more intrigued in the research problem.

4.4 Improving Baseline Models by Leveraging Expert Reviews

The moderate preliminary results led us to think of better ways to explore the linkages between noisy and curated domains. Hence, logically the next path was to transform noisy domain to be more like the curated one by just considering reviews that are similar to the curated domain review.

Now, our goal is to identify reviews that play a major role in determining product as *wirecutter_selected* or *wirecutter_unselected*, and to inspect such reviews to see if they have similar characteristics as the Wirecutter review. Reading only such reviews would help consumer to make better perspective about a product rather than by reading top most helpful reviews, as review helpfulness is merely a measure of how helpful a particular review is and not how "good" the reviewed product is. We will further examine whether using only expert reviews increases accuracy for the prediction task stated earlier.

4.4.1 Discovering Expert Reviews from Crowd

Although Amazon receives all of the product reviews from general consumers (crowd), there exist some reviews that are exceptionally well-written, informative and helpful for other users. Our next task is identifying such reviews (we dub such reviews as *Expert reviews*). So, for every product, we segregate all reviews in 2 classes:

1. Expert reviews (The reviews that align with Wirecutter review).

2. Other noisy reviews.

Wirecutter provide hand-made highly detailed reviews on various products across categories. These reviews are made through vigorous reporting, interviewing, and testing by teams of journalists, scientists, and researchers. The review writers and editors are never made aware of which companies may have established affiliate relationships with Wirecutters business team prior to making their picks. Hence, they claim their curated review to be un-biased [32]. Hence, the proposed *Expert reviews* are also claimed as un-biased owing to the fact that they align with the Wirecutter review. Wirecutter.com segregates their curated review into various sections [33]. Some of the sections are as follows:

- **Why should you trust us?** : This section goes over the justification of why the author qualifies as an *Expert* in that particular category. Wirecutter also iteratively updates their review, so this section gets updated along with it.

- **Features to consider** : Vital category-dependant features to consider while buying a product from this category are discussed in this section.

- **How we picked and tested** : This section goes over the vigorous testing in numerous settings, done by the authors. They refer and cite various online blogs/resources that helped to form perspectives about products. They generally discuss all the technical specifications of products here, and compare them with those of other products.

- **Flaws but not dealbreakers** : This section consists of minor flaws of the Wirecutter recommended products. They also justify why these flaws are not dealbreakers to eliminate the product from their recommendation, comparing it with other products.

- **Who else likes it** : This section provides links to various articles and websites that like the products selected by Wirecutter, along with a small summary of that article itself.

- **What to look forward to** : This section mentions any new competitive products that might be released soon into the market.

- **The competition** : All the products taken into consideration by Wirecutter in that category are listed in this section.

- **Footnotes** : Authors describe their final thoughts in this section.

- **Sources** : This is the list of all the references considered by authors while wring the whole review.

We dub this whole curated description as *Wirecutter description* for our future references. We derive *Expert reviews* from Amazon.com comparing it with *Wirecutter description*. Further, we tried to explore if the use of these *Expert reviews* solely, improve our models.

The first task was to separate *Expert reviews* from the crowd. This itself being a non-trivial task, there are many ways to do it. For the purpose of this task, we assumed approximately 30% of all the Amazon reviews to be *Expert reviews*. This percent number was decided by extensively checking the results on varying the threshold. We decided to explore the following two approaches, just to see if the use of *Expert reviews* solely could improve our models:

1. Calculating the Jaccard index between wirecutter description and the review, then thresholding per category. This method yielded unsatisfactory results as it suffers from few or no overlapping items.

2. Calculating the cosine similarity between the tf-idf vectors of Wirecutter description and the review, then thresholding per category. Cosine similarity measure signifies how similar two documents are likely to be in terms of their subject matter [34]. This, coupled with its low complexity, led us to adopt cosine similarity for all future models.

This method captures the wisdom of crowd and domain experts.

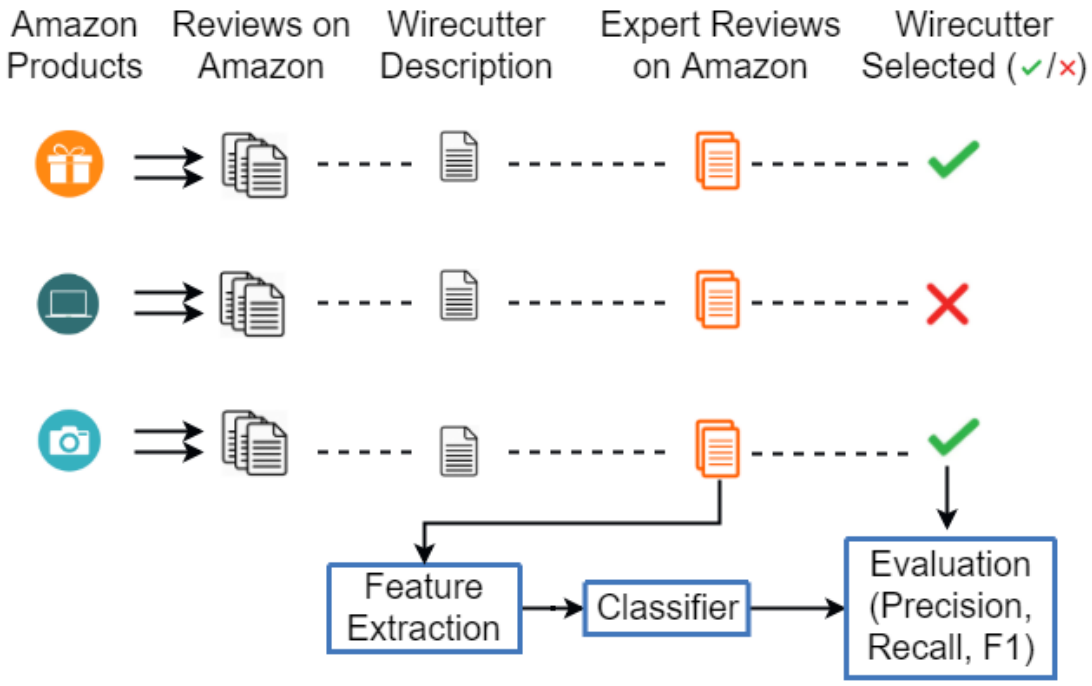


Figure 4.6: General Setup for Model with Expert Reviews.

Figure 4.6 shows the general setup for models with expert reviews on a category.

4.4.2 Meta-Features Model on Expert Reviews (MF-ER)

With the set of *Expert reviews*, we calculated features on per product basis, as done earlier. We created models with settings similar to the ones we used earlier. The meta-feature model consisted on features from the star-ratings such as the average of ratings, median, absolute deviation from mean and variance,etc [27]. The features also entail average of helpful votes, review body length, review title length and review post latency (time elapsed between product launch and posting of a particular review) for all reviews of a product. All the models described further use the same Random Forest Classifier with 5-Fold Cross-Validation scheme. Table 4.4 shows results from the meta-feature model.

Categories	Precision	Recall	F1 Score
Headphones	0.59	0.38	**0.46**
Cameras	0.54	0.44	**0.48**
Laptops	0.49	0.45	**0.47**
Projectors	0.48	0.33	**0.39**

Table 4.4: Classification Results using Random Forest Classifier with Meta-Features of *Expert reviews* and 5-Fold Cross-Validation Scheme

Due to the noise removal, we witnessed comparative increase in the results. This got us the inspiration that we are on the right track for this research.

4.4.3 Context based Features Model on Expert Reviews (CBF-ER)

Then, for textual features, term frequencyinverse document frequency were calculated by combining the selected reviews for each product, and these vectors were used as features for model. Table 4.5 shows results from the tf-idf features model.

A slight improvement is seen in the results.

Categories	Precision	Recall	F1 Score
Headphones	0.52	0.25	**0.34**
Cameras	0.49	0.41	**0.45**
Laptops	0.44	0.36	**0.40**
Projectors	0.31	0.27	**0.29**

Table 4.5: Classification Results using Random Forest Classifier with tf-idf Features of *Expert reviews* and 5-Fold Cross-Validation Scheme

4.4.4 Word2Vec Embeddings Model on Expert Reviews

Now, to fuse semantic and syntactic features into our model, we use 'word2vec' embeddings. After calculating the word embeddings, we combined them to form product-embeddings using three ways.

1. Averaging all the word embeddings for given product. Results are shown in table 4.6

Categories	Precision	Recall	F1 Score
Headphones	0.49	0.43	**0.46**
Cameras	0.36	0.33	**0.34**
Laptops	0.39	0.29	**0.33**
Projectors	0.44	0.48	**0.46**

Table 4.6: Classification Results using Random Forest Classifier with Averaged Word Embedding Features of *Expert reviews* and 5-Fold Cross-Validation Scheme

2. Considering embeddings of only top 50 chi-squared most important topics, then averaging them. Results shown in table 4.7 imply further increase in F1 scores.

3. In this model (W2V-ER), we average word embeddings in reviews to form review embeddings. These review embeddings are then weighted by cosine

Categories	Precision	Recall	F1 Score
Headphones	0.45	0.48	**0.46**
Cameras	0.54	0.45	**0.49**
Laptops	0.46	0.39	**0.42**
Projectors	0.51	0.43	**0.47**

Table 4.7: Classification Results using Random Forest Classifier with Top 50 chi-squared Topic's Averaged Embeddings from *Expert reviews* and 5-Fold Cross-Validation Scheme

similarity between tf-idf vectors of the review and wirecutter description. For product embeddings, simple average of these weighted embeddings is taken. These product embeddings are further used as product feature vectors in this model. As seen in table 4.8, this method yield the best results.

Categories	Precision	Recall	F1 Score
Headphones	0.75	0.61	**0.67**
Cameras	0.62	0.55	**0.58**
Laptops	0.68	0.63	**0.65**
Projectors	0.71	0.59	**0.64**

Table 4.8: Classification Results using Random Forest Classifier and weighted average of the text embeddings by the cosine similarity to *wirecutter description* and 5-Fold Cross-Validation Scheme

Classification using our last proposed method shows an interesting trend. The performance metric shows a drastic increase as compared to all the other classifiers. This can be explained as we are giving more weight (more importance) to reviews those are more similar to *Wirecutter description*.

To gain a clear perspective of the results, we manually analyzed the miss-classified products and found that most of the miss-classified products had insufficient *Expert*

reviews after filtering the noisy reviews. Hence, with proper tuning of the thresholding value while filtering the noisy reviews, results can be improved at a greater extent.

4.5 Analysis of Results

We now initiate the analysis of our results, plotted in Figure 4.7. We see the most variation of F1 scores in *headphones* category. This is the biggest category in terms of product reviews. As seen from Figure 4.3, its also the most imbalanced category of all products-wise (only 69 products were selected from 602 products). This major imbalance accounts for the major variation in the results of different models. Consequently, similar logic could be extrapolated towards *cameras* category, which is almost balanced category (115 products selected from 281 products), hence the less variation in results.

From Figure 4.7 its also clear that the model *W2V-ER* performs better in *headphones* category owing to the huge corpus it gets trained on. To analyze the comparatively better results from our expert-reviews-based models, we tried explore what qualities of reviews make them classify as *expert reviews*.

We plot histograms of review lengths across all reviews and *expert reviews* separately for all categories as seen in Figure 4.8. Histogram for *expert reviews* is a right-shifted and scaled version of the histogram from all reviews in most categories. This implies the *expert reviews* being more lengthy than most reviews. We indicate average review length on the plots and observe *expert reviews* to be more bulkier than average length. This observation was solely based on plots generated, and hence its not spurious.

Some other observations:

- W2V model on all reviews result in inferior results as compared to W2V-ER,

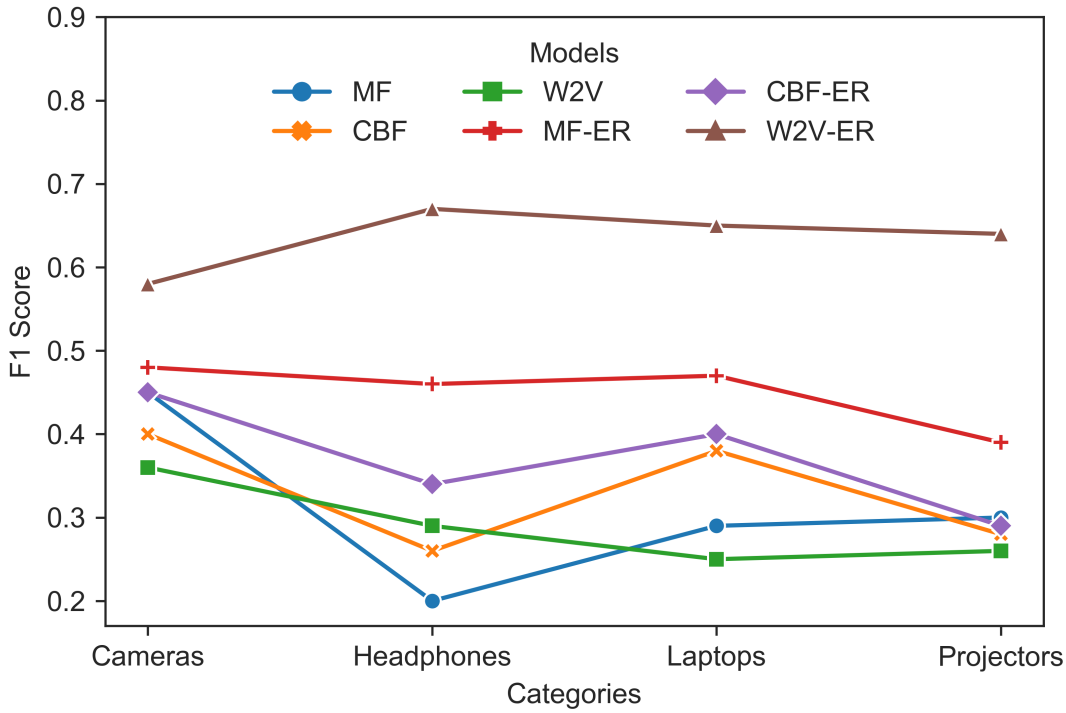
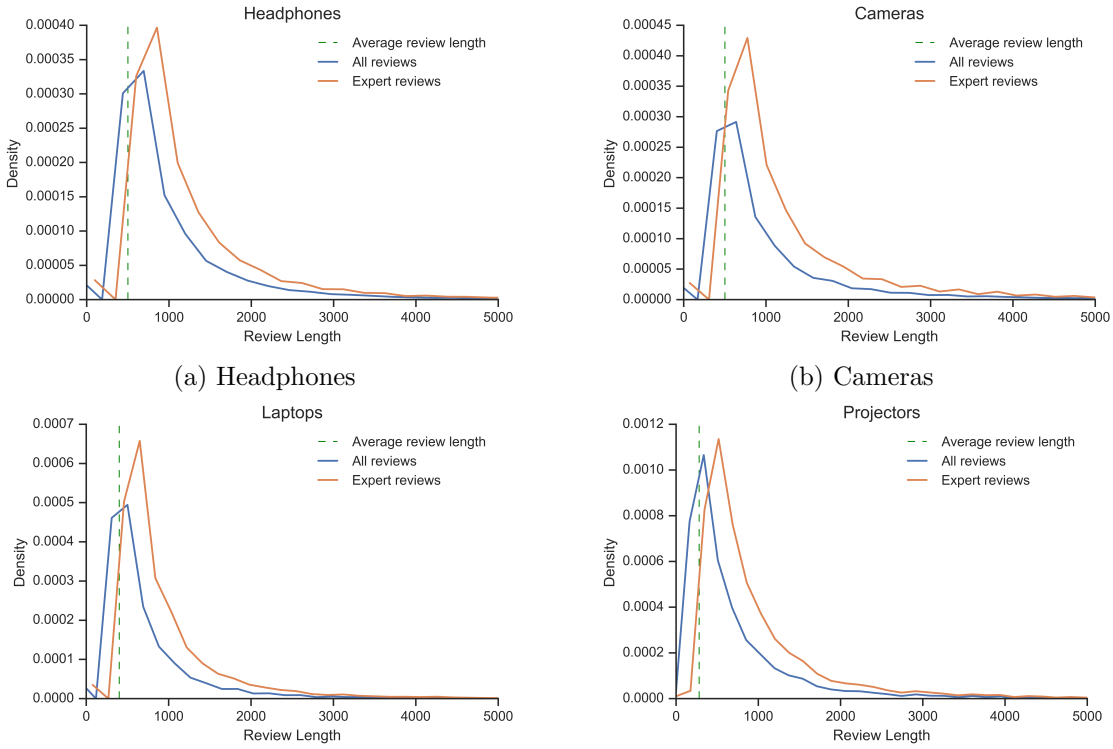


Figure 4.7: F1 Scores for all Models by Category.

on all categories. This is attributed to learning qualitative embedding due to removal of noisy context. In other words, Word2vec comparatively learns better latent feature embeddings being trained on high quality data (Expert reviews), than when its trained on all reviews.

- General notion is that the number of reviews per product is positively correlated with goodness of product, which is disproved here. *Headphones* is the biggest category with highest reviews per product and *Projectors* is smallest category with lowest review per product. Both categories yield similar results for W2V-ER, this suggests that number of reviews per product does not determine the goodness of that product.

- There is not significant increase from CBF to CBF-ER, contrary to expectation.



(a) Headphones

(b) Cameras

(c) Laptops

(d) Projectors

Figure 4.8: Review Length Histograms over all and Expert Reviews across Categories

This might be due to the fact that tf-idf ignores semantics and only considers relative occurrences of words between documents. Filtering expert reviews out of crowd might not have changed relative occurrences of words significantly.

- Expert reviews identified by our proposed strategy are generally longer in length than average length of all reviews. They capture the wisdom of crowd and domain experts, hence they are better representative of how good the product is in an unbiased way.

- Highly imbalanced categories in terms of instances per class yield results with huge variance and vice versa.

After exploring *Expert reviews* to greater extent, we observed that they had one

or more attributes of the following:

- Opinionated: Gives a clear and strong opinion about product coupled with less ambiguities towards product
- Detailed: More in-depth description is provided. Such reviews are often informative due to its rich content and often constitute better insights regarding pros/cons and comparisons to other products.
- Uniquely written: These reviews often posses unique writing styles and are fluent and easy to understand. They are rich in language structures along with good insights
- Written by domain experts: These reviews are written by domain experts themselves or those consumers who used the product for a long time.

After observing *expert reviews*, we tried to explore the reasons why *W2V-ER* model performed far better than others. In *W2V-ER* model, we up-weight the review embeddings by the cosine similarity between tf-idf vectors of the review and Wirecutter description. This setup gives more relevance to reviews that align vastly with Wirecutter description in the embedding space. Thus, these reviews are far more representative of goodness of the product. Hence, we observe the sudden rise in results.

5. CONCLUSIONS AND FUTURE WORK

In our research, we studied various approaches to link noisy and curated domains. We started off with basic analyses to see if there indeed are any links between both domains. Restricted by a good data source, we created our own data-set using web scraping and built different classification models on top of it. Thus we identified a challenging problem (linking crowd to curated reviews) and contributed a new dataset. The preliminary results led us to believe that the problem statement we were dealing with is a hard and non-trivial problem indeed. This thought process got us even more intrigued in the research problem.

We indicated existence of unbiased and highly informative reviews in noisy review domains and proposed a way to identify them (Expert reviews). We also proposed best products recommender engine that provides value to the general public without any personalized inputs. We proved that the likelihood of discovering best products from a category can be significantly improved by leveraging expert review features solely.

Further, we will analyze how the identified expert review features determine good products in other curated review domains, except Wirecutter. Even the process of expert reviews identification could be improved by including review data from various curated review websites, as this would capture the wisdom of multiple domain experts via multiple curated domains. Future work may also include analyzing other crowd-sourced websites to see if similar results hold in those domains as well.

We are currently only considering products that are considered worthy of hand-curated review by Wirecutter, they constitute products under their 'The competition' and 'Our picks' sections for our analysis. Going ahead, we would explore all the

products on Amazon under a category and see if we could mine good products out of them.

We proposed best products recommender engine that provides value to the general public without any personalized inputs. But in future we would like to transform this model to recommend personalized products by leveraging user history and profiling every product and user.

Currently our model uses the curated review for identifying *Expert reviews* from the crowd. In future, we will works towards identifying *Expert reviews* without the use of curated review. This will facilitate moving from noisy domains to curated ones without human interaction, essentially making them easy to scale.

REFERENCES

[1] Amazon.com, Inc., the e-commerce website. Retrieved on September 24, 2018 from https://www.amazon.com/.

[2] The Wirecutter, A New York Times Company. Retrieved on September 24, 2018 from https://thewirecutter.com/.

[3] Skymind, Artificial Intelligence Wiki. Retrieved on September 24, 2018 from https://skymind.ai/wiki/word2vec.

[4] K. Z. Zhang and M. Benyoucef, "Consumer behavior in social commerce: A literature review," *Decision Support Systems*, vol. 86, pp. 95 – 108, 2016.

[5] M. Farber, "Consumers are now doing most of their shopping online." `http://fortune.com/2016/06/08/online-shopping-increases/`, 2016.

[6] N. Hu, L. Liu, and J. J. Zhang, "Do online reviews affect product sales? the role of reviewer characteristics and temporal effects," *Information Technology and Management*, vol. 9, pp. 201–214, Sep 2008.

[7] P. F. Wu, "In search of negativity bias: An empirical study of perceived helpfulness of online reviews," *Psychology & Marketing*, vol. 30, no. 11, pp. 971–984, 2013.

[8] C. Tan, E. H. Chi, D. Huffaker, G. Kossinets, and A. J. Smola, "Instant foodie: predicting expert ratings from grassroots," in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, CIKM '13, (New York, NY, USA), pp. 1127–1136, ACM, 2013.

[9] M. C. Surabhi, "Natural language processing future," in *2013 International Conference on Optical Imaging Sensor and Security (ICOSS)*, pp. 1–3, July 2013.

[10] B. Li, Q. Yang, and X. Xue, "Transfer learning for collaborative filtering via a rating-matrix generative model," *ACM*, pp. 617–624, 2009.

[11] W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang, "Transfer learning in collaborative filtering for sparsity reduction," *AAAI*, pp. 230–235, 2010.

[12] Y. Zhang, B. Cao, and D. Yeung, "Multi-domain collaborative filtering," *CoRR*, vol. abs/1203.3535, 2012.

[13] S. Xie, Q. Hu, J. Zhang, J. Gao, W. Fan, and P. S. Yu, "Robust crowd bias correction via dual knowledge transfer from multiple overlapping sources," in *2015 IEEE International Conference on Big Data (Big Data)*, pp. 815–820, Oct 2015.

[14] H.-J. Min and J. C. Park, "Identifying helpful reviews based on customers mentions about experiences," *Expert Systems with Applications*, vol. 39, no. 15, pp. 11830 – 11838, 2012.

[15] N. Chen, J. Lin, S. C. H. Hoi, X. Xiao, and B. Zhang, "Ar-miner: Mining informative reviews for developers from mobile app marketplace," in *Proceedings of the 36th International Conference on Software Engineering*, ICSE 2014, (New York, NY, USA), pp. 767–778, ACM, 2014.

[16] Q. Cao, W. Duan, and Q. Gan, "Exploring determinants of voting for the helpfulness of online user reviews: A text mining approach," *Decision Support Systems*, vol. 50, no. 2, pp. 511 – 521, 2011.

[17] J. Otterbacher, "'helpfulness' in online communities: A measure of message quality," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, (New York, NY, USA), pp. 955–964, ACM, 2009.

[18] N. Korfiatis, E. Garca-Bariocanal, and S. Snchez-Alonso, "Evaluating content quality and helpfulness of online product reviews: The interplay of review helpfulness vs. review content," *Electronic Commerce Research and Applications*, vol. 11, no. 3, pp. 205 – 217, 2012.

[19] M. Kokkodis, "Learning from positive and unlabeled amazon reviews: Towards identifying trustworthy reviewers," in *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, (New York, NY, USA), pp. 545–546, ACM, 2012.

[20] Y. Liu, J. Jin, P. Ji, J. A. Harding, and R. Y. Fung, "Identifying helpful online reviews: A product designers perspective," *Computer-Aided Design*, vol. 45, no. 2, pp. 180 – 194, 2013. Solid and Physical Modeling 2012.

[21] R. Mitchell, *Web Scraping with Python: Collecting Data from the Modern Web*. O'Reilly Media, Inc., 1st ed., 2015.

[22] D. Glez-Pea, A. Lourenco, H. Lpez-Fernndez, M. Reboiro-Jato, and F. Fdez-Riverola, "Web scraping technologies in an api world," vol. 15, 04 2013.

[23] E. Vargiu and M. Urru, "Exploiting web scraping in a collaborative filtering-based approach to web advertising," *Artif. Intell. Research*, vol. 2, pp. 44–54, 2013.

[24] A. Mehlfhrer, "Web scraping: A tool evaluation," 2009.

[25] A. Finn, N. Kushmerick, and B. Smyth, "Fact or fiction: Content classification for digital libraries," 07 2001.

[26] S. Gupta, G. Kaiser, D. Neistadt, and P. Grimm, "Dom-based content extraction of html documents," in *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, (New York, NY, USA), pp. 207–214, ACM, 2003.

[27] C. Danescu-Niculescu-Mizil, G. Kossinets, J. Kleinberg, and L. Lee, "How opinions are received by online communities: A case study on amazon.com helpfulness votes," in *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, (New York, NY, USA), pp. 141–150, ACM, 2009.

[28] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *CoRR*, vol. abs/1310.4546, 2013.

[29] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[30] T. M. Sanjeev Arora, Yingyu Liang, "A simple but tough-to-beat baseline for sentence embeddings," *ICLR*, 2017.

[31] R. Socher, "Cs224d: Deep learning for natural language processing: Lecture 02,"

[32] J. Cheng and B. Lam, "Wirecutter: About us."

[33] J. Cheng and B. Lam, "wirecutter.com."

[34] A. Singhal and I. Google, "Modern information retrieval: A brief overview," vol. 24, 01 2001.