Title
SweetSpotter: Mining Consumer Reviews For Consumer Packaged Goods Product Optimization

Author
Sun, Xing

Publication Date
2021

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

SweetSpotter:
Mining Consumer Reviews
For Consumer Packaged Goods
Product Optimization

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Applied Statistics

by

Xing Sun

2021

ABSTRACT OF THE THESIS

SweetSpotter:
Mining Consumer Reviews
For Consumer Packaged Goods
Product Optimization

by

Xing Sun
Master of Applied Statistics
University of California, Los Angeles, 2021
Professor Ying Nian Wu, Chair

An end-to-end pipeline of text mining over consumer reviews on Amazon under the search phrase "face mask" was built for the product designers, engineers and manufacturers to identify "sweet spots" for product engineering and manufacturing optimization. Compared with current Natural Language Processing (NLP) and in particular Aspect Sentiment Classification (ASC) approaches, this research achieved state-of-the-art (SOTA) classification accuracy in 31 classes of aspects (0.83 accuracy) and 3 classes of sentiments (0.91 accuracy), with a small (<1,500) training dataset. The SweetSpotter pipeline took in raw review texts scraped from Amazon, split them into minimal semantic units, fine-tuned on bert-base-uncased transformer with training dataset labeled by a human expert, classified nearly 400,000 text units, and delivered insights on the most impactful and meaningful features to improve the product in terms of user experience. It turns out that consumers care most about fit, least about look, on face masks.

The thesis of Xing Sun is approved.

Hongquan Xu

Mark Handcock

Brett Hollenbeck

Akram Mousa Almohalwas

Ying Nian Wu, Committee Chair

University of California, Los Angeles

2021

For my preteen son . . .

who supported a graduate schooling mom

Contents

List of Figures

List of Tables

Preface

I am grateful for the professors on my committee, especially my chair **Professor Wu** who guided me along the path of this fascinating field in Machine Learning. I am indebted to PhD students: **Liang Qiu** who generously shared his knowledge and know-how in Natural Language Processing and **Jake Elmstedt** who inspired me on Amazon scraping.

The technologies I used to build the SweetSpotter pipeline epitomize the *Open Source* movement and philosophy started in the software industry. I could not have built this powerful and practical process within the time constraint of a full-time working graduate student without **Amazon** for cataloging the review data, **R** and **Python** for processing or visualizing data, **Huggingface** and **PyTorch** for Machine Learning engines, and **Google Colab** for cloud computing. **Google Scholar**'s automated citation formatting, **Overleaf LaTex**'s programmed thesis formatting and **Louis Yang**'s UCLA thesis template were all huge time savers.

CHAPTER 1

Introduction

User Generated Content (UGC) on social media, in the format of texts, photos, audios, videos, or live streams to name a few, empowers individuals to participate in a global conversation on a topic. While massive "digital exhaust" is constantly generated on ever-expanding online platforms, the assemblage of structured UGC can be a salient source of "partial truth" to gain insights from the collective human knowledge in a domain. Consumer product review is one such topic-specific UGC venue, where people share their individualized user experience on a product with the world, published under a unique product identifier, sortable by popularity, recency or star rating. Amazon is the best place to access reviews for Consumer Packaged Goods (CPG). In recent years, researchers have approached text mining on consumer reviews in myriad ways.

This paper endeavors to build a "4C" pipeline of collecting, cutting, classifying, compiling CPG reviews which delivers hierarchically categorized product features in positive, neutral or negative sentiments to show the big picture of how collectively consumers value and evaluate predefined attributes on a product, for the benefit of the product designing, engineering, and manufacturing community, whose mission is to perfect their crafts in bringing the next generation products to consumers by making the optimal decisions of allocating engineering and manufacturing resources to the most impactful product attributes. Finding the "sweet spots" through the 4C CPG product review NLP pipeline is the purpose of this thesis.

1.1 Natural Language Processing

Natural Language Processing as an interdisciplinary research field at the intersection of computer science, information science and linguistics has been evolving fast in recent years when the breakthroughs in Language Modeling such as transformer in 2017[50], BERT in 2019[17] and GPT in 2020[6] paved new paths to process languages with their promises and limits[44]. Consumer product reviews from multiple product listings and review pages require Multi-Document Summarization (MDS). There are two major approaches to summarization: extractive and abstractive. While abstractive summarization generates new texts based on feed-in multi-document texts, extractive summarization parses and classifies texts. For consumer reviews that contain sentiments about aspects of a product, one focus area of extractive summarization is Aspect Sentiment Classification (ASC), also known as Aspect-Based Sentiment Analysis (ABSA).

Below is the "family tree" to show where the SweetSpotter pipeline is positioned in the NLP lineage.

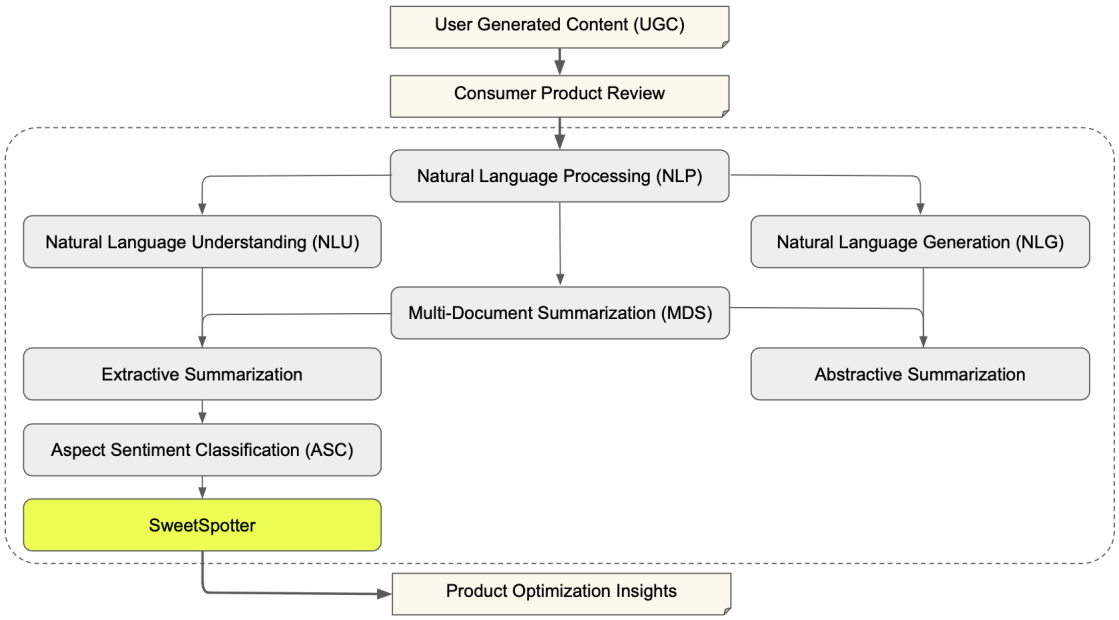

Figure 1.1: Natural Language Processing Lineage

1.2 Summarization Methods and Metrics

Since 2018, abstractive summarization is flourishing with diverse approaches, such as Diverse Beam Search[14,51], OpinionDigest[46], MeanSum[13], Unsupervised[8,15], Unsupervised with Denoising[34], Unsupervised with Tree-Structured Topic Guidance[28], BART[2], Convex Aggregation[27], PEGASUS[58], and PASS[41] to better capture contents per the coherency, fluency, and non-redundancy criteria laid out in DUC 2005[16]. Performance metrics for abstractive summarization, such as ROUGE[35,40], BERTScore[60], BLEU[42], SummEval[19], METEOR[32], to evaluate abstractive summarization were devised simultaneously. While abstractive summarization was proposed for generating reviews for cold products (products without any reviews)[43], critiques of abstractive summarization centered around faithfulness[11,37].

Extractive summarization came a long way from "bag of words" and LSTM[23], Topic Modeling[47], graphic or fuzzy logic based[45] days to Support Vector[29], Vector-Quantized Variational Autoencoder (VQ-VAE)[3], Unsupervised[21], Weakly Supervised[4], and Automatic Clustering[48] variations. GLUE[52] is a major benchmark, besides precision, recall and F1 performance metrics.

A niche of extractive summarization is to extract aspects (aka features or attributes) and sentiments (aka opinions or polarity) from texts, as demonstrated in recent works with Convolutional Neural Network (CNN)[55] and clustering[30] methods.

1.3 Machine Learning Limitations

While NLP is progressing exuberantly with Machine Learning (ML) technologies, it takes more than computational excellence to solve natural language problems, because language is a long-evolved operational function of human existence, deeply embedded with human activities, experiences and contexts. Researchers[31] attributed the limitation of current NLP methods to "current research setup, which involved uncurated, automatically collected datasets and non-informative evaluations protocols." In other words, there's a divide between the human world and the computational world. The latest AI100 whitepaper[39] pegged the Artificial

Intelligence (AI) limitation onto "Common Sense", which is about "general intelligence" that humans operate on without being explicitly aware of it, "including a vast amount of mostly unconscious knowledge about the world, an understanding of causality ... and an ability to perceive abstract similarities between situations, that is to make analogies."

Another bottleneck on NLP is its heavy reliance on a costly labeled dataset. Several methods were developed to recycle labeled datasets and expand it with unlabeled datasets, including Few-Shot Learning[9,10], Snippext[38], Content Planning[1], and Label Bootstrapping[57].

1.4 Fake Review

As organized UGC on a uniquely identified product, product reviews gained so much popularity and influence since Amazon perfected user review soliciting, sorting, and sharing process to make itself an essential user-seller interaction platform on its regional and multilingual sites that some products are artificially rated with overblown star rating and overly positive reviews, called "fake reviews" to sway consumer purchase decisions. A 2016 study[5] showed that products with reviews increase purchase conversion rate by 270% than cold products. Benign fake reviews can be a beguiled marketing tool to highlight the benefits and selling points of the product with seller-sponsored UGC[25] so effective that there exists a market for underground fake review transactions[22]. Malicious fake reviews can trick unwitting consumers to buy products lauded by misrepresented reviews. Fake review detection technology such as Fakespot and ReviewMeta for consumer protection is still rudimentary in its algorithms. The prevalence of fake reviews on e-commerce sites testifies the influence of reviews on consumer decision making.

1.5 Review Summarization

While some researches focused on sentiment-only review text mining in different use case applications, such as user preference[33] and business analysis[18], and with different methods, such as SenBERT-CNN[54] and Natural Language Toolkit (NLK)[53], many researchers mined

product reviews on two dimensions: aspect and sentiment, as early as 2004[26]. To get the most useful insights from product review texts, researchers utilized the "helpfulness" vote to rank extracted reviews[56], RoBERTa-large to classify key points[7], and Sentence Transformer Embedding to identify helpful sentences[20]. Researchers also devised summarization pipelines for the benefit of both businesses with SWOT (Strength, Weakness, Opportunities and Threat) analysis[12] and consumers with hierarchical aspect extraction[36]. In consumer insights mining, sentences with tips on how to fix or repurpose the product account for "only 4.52% of all labeled sentences"[24] and innovation ideas from consumers defined per a research[59] takes up only "0.21% of all sentences". To help distill consumer needs from large corpora, researchers also used the machine-human hybrid method to "eliminate irrelevant and redundant content"[49]. In this paper, redundancy is used to evaluate the product attribute relevance. The more mentioned and redundant, the more relevant a product attribute is to the collective consumer experience.

CHAPTER 2

SweetSpotter

When continued advances of the Machine Learning-powered NLP methods in product review mining are to be carried out by many researchers for numerous years to come, the SweetSpotter pipeline addressed some issues in a simple streamlined structure and delivered useful consumer insights on product features with SOTA accuracy, taking into account of the contextual understanding of product engineering, product review and linguistic expressions before translating the business use case to a Machine Learning application.

The unique contributions of SweetSpotter are demonstrated in the below aspects:

- built by and for CPG product designing/engineering/manufacturing professionals
- segmented effectively to minimal semantic units for better ML performance
- delivered with actionable Business Intelligence (BI) for Product Optimization

2.1 Product Engineering

The Industrial Design framework is based upon user experience. How users interact with a product is what product designers engineer around. Dimensions like ergonomics, aesthetics and economics are key factors in the shaping of the final products. Designers, engineers and manufacturers constantly face the challenges to design a winning product with all attributes configured optimally. Product designers specialize in product looks, while product engineers in product functions. Sometimes, the two roles merge into one position. But designers or engineers are limited to their experience and imagination of how their products are going to be used and experienced by consumers. Industry common practices such as lab testing or

consumer direct feedback address such limitations to an extent. It is still limited to a small sampling of the diverse user experience. Online consumer reviews can expand this narrow bandwidth of consumption to conception feedback dramatically, with the right tool.

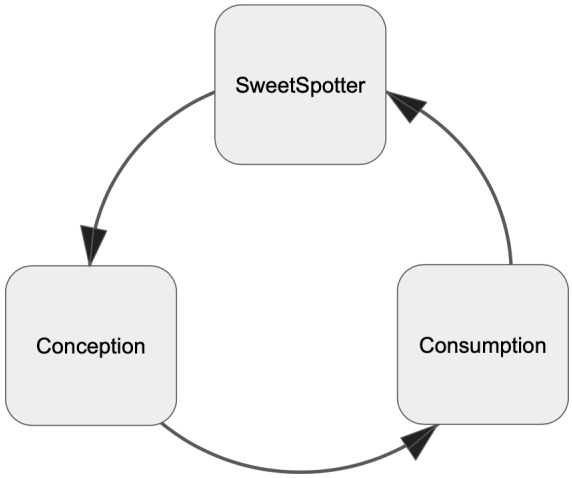

Figure 2.1: Consumption-Conception Loop

SweetSpotter is this missing link on the feedback loop from consumption to conception. Otherwise, it is a one-way flow from conception to consumption, which can cost both consumers and businesses when a poorly engineered product wastes the business' reputation and consumer's money. Once the product is conceptualized, engineered, manufactured, rolled off the production line and delivered through an intricate distribution and logistic network, the receiving end of the product is the consumer using it in a particular setting, such as weather, occasion, and purpose (sometimes consumers repurpose the product). The wear-and-tear and the expectations of the product are going to vary accordingly. Consumer product reviews on e-commerce sites like Amazon is the perfect place to mine collective consumer feedbacks and turn them into business intelligence, for the next season or generation product conception.

SweetSpotter constructed the product aspects classifier based on the product engineering perspective, which grouped all feature-specific reviews into five major categories: color, fit, material, manufacturing and user experience. The first three umbrella terms are the three fundamental industrial design elements, which appeal to consumers' visual, spatial and tactile

senses. Depending on the use of the product, the values on each of these three senses are going to be prioritized differently. This thesis is going to find out the value proposition of the three senses on the example of "face mask".

2.2 Product Review

Who are the reviewers? Not every customer leaves a review or as many reviews as others. By the "open mic" setup, the reviews written and published by a subset of consumers are going to be biased. Before we even use the data from consumer reviews, it is better to keep in mind its limitations.

The motive of customers spending time typing up and publishing reviews can be altruistic, like sending warnings to future buyers. Or they want to vent or express their strong opinions on an "open mic". Or they want to build their credibility and become a known product reviewer with a following and eventually as any influencer on Social Media does produce contents with advertising embedded messages. One metric for fake review detectors to use is the Review Count by a reviewer. If the review count published by one reviewer is too high ($>$ 15 reviews), it is called "overrepresented participation" by ReviewMeta and treated as a flag for a fake review.

Figure 2.2 is the distribution of overall star ratings on all 1078 collected product review postings. The median is 4.6 and the mean 4.48 on the 1 to 5 scale. One possible explanation is that low-rating products simply won't survive long on a digitally-flattened open marketplace. So by the time when the consumer reviews on "face mask" were collected in July 2021, the postings on Amazon are mostly 4.0+ rated products by "natural selection".

With product review built-in biases, the review sentiment reading would be most likely skewed toward positive, which is confirmed by this research. Overall general product reviews are 76% positive, 21% negative, and 3% neutral. Aggregated review sentiments across all aspects are 64% positive, 31% negative, and 5% neutral. It's worth noting that when people are specific about product features, it is not as positive as general comments.

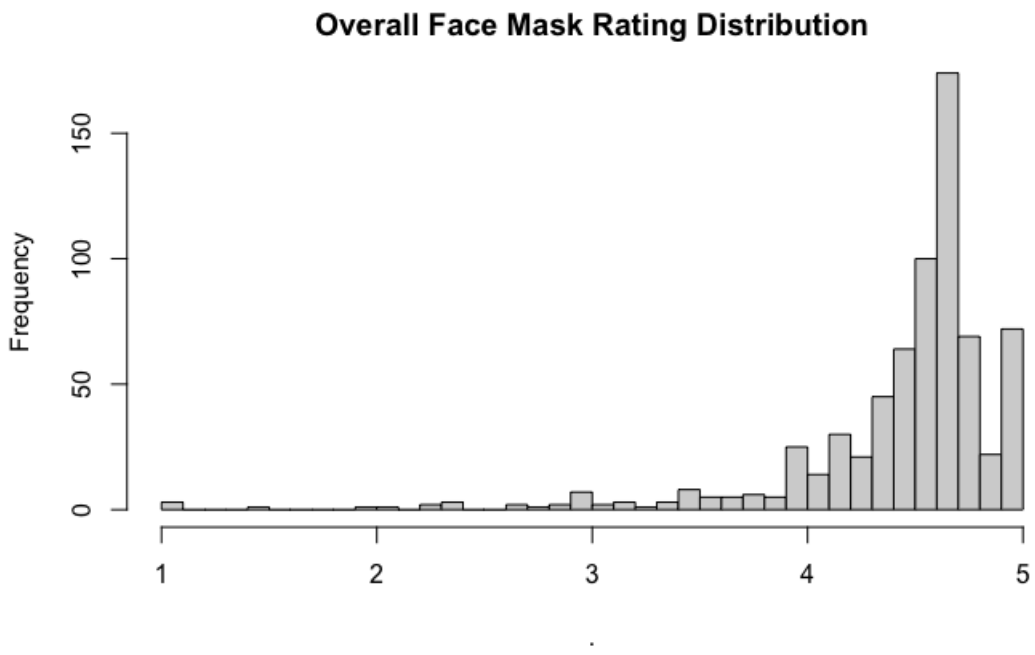

Figure 2.2: Overall Product Rating Distribution

2.3 Natural vs Machine Language

The SweetSpotter pipeline addressed the human context issue with a carefully curated training dataset and minimal raw text modifications by commonly used text mining techniques, such as lowering cases or removing stop words or correcting typos, in trust of the SOTA Language Model BERT with its bi-directional attention mechanism adeptly tuned to reflect the implicit connections in human experience, namely user-product interaction in this study.

When labeling product aspects, the training dataset creator would adjust the labels with the context provided within that semantic unit without human inference from previous text unit under the same review as well as make it explicit human world cross-references. For example, when labeling input text “are great for acne”, humans can tell this positive sentiment is for the mask “breathability”. A machine cannot infer that acne-breathability connection. That is what the training dataset is for, to manually provide such non-linguistic human contextual connections. The classification result with "acne" in the texts are shown on Table 2.1.

Classified Texts with the Word *acne*

Review Text	**Aspect**	**Sentiment**
my acne is going away from wearing this mask	breathability	negative
caused the WORST “mask-ne” (mask acne) ever	breathability	negative
are great for acne	breathability	positive
if you’re a fan of mask acne these are PERFECT...	breathability	positive
Haven’t broken out in acne	breathability	positive
and have no mask acne anymore	breathability	negative
doesn’t give u acne	breathability	positive
my face breaks out with acne from irritation	breathability	negative
(I never have acne)	breathability	positive

Table 2.1: Example Texts Classification Result

While fine-tuned BERT with this information can recognize "acne" related to "breathability" and classified correctly for the most part, the machine is limited at understanding sarcasm. The text "if you’re a fan of mask acne these are PERFECT" is wrongly classified as positive. English is notorious for its prevalence of sarcasm, perceived as a culturally applauded trait of the language user. Machine often takes words at their literal level, unless it can be specifically trained with sarcastic use cases. Given the limited resources with this thesis, the sarcasm misreading is not addressed, but in the analysis, we can take the ML reading of the texts with a grain of salt.

2.4 SweetSpotter Pipeline

SweetSpotter 4C pipeline as illustrated in Figure 2.3 takes in raw texts scraped from Amazon reviews and outputs a classified aspect-sentiment result which can be further analyzed and visualized for the benefit of product optimization.

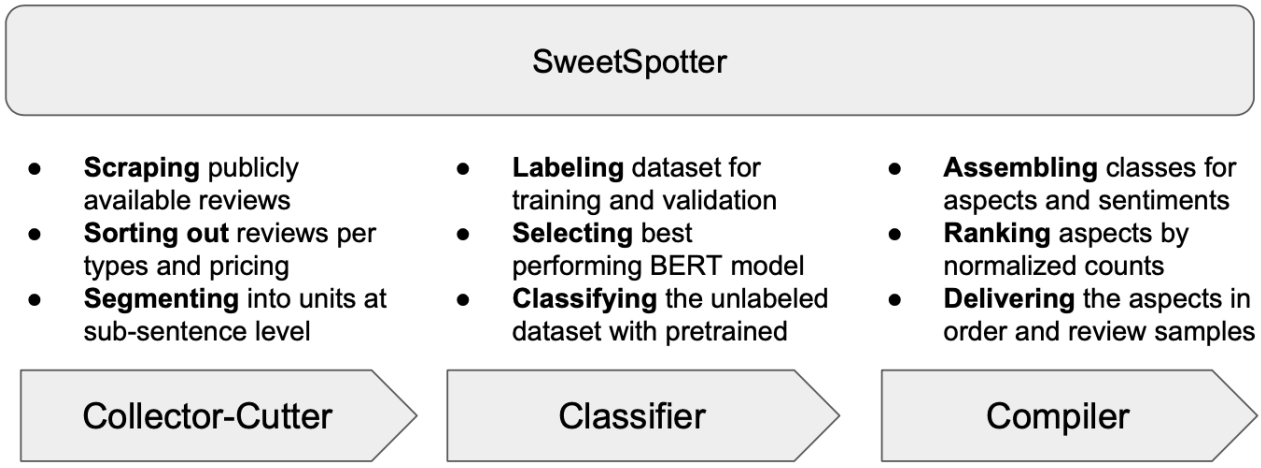

Figure 2.3: SweetSpotter Pipeline

2.4.1 Collector

Crawler coded in R with RSelenium package scraped review data in two steps:

Step 1: scrape all product listings on a search result page, under the search word "face mask", and loop to the last page. The maximum depth for any product search word listing is 306. Several search result sorting orders were used to get to a total of 1078 unique product listings after deduplication.

Step 2: scrape till the last review on the last page of all reviews under each of the unique product listings. 78,191 review documents were saved for the training dataset and X dataset creation, after filtering out non-pandemic-protection-use product listings.

After scraping, SweetSpotter gathered all scraped texts, one document per product listing, into a depository before minimal "cleaning": filter out any product listing under "face mask" but is a cosmetic skincare product or face protection for sports, which normally can be weeded out by the unit price of the product. After that, cherry-pick any products that are not the product of interest. For a future project, this can be automated based on product attribute clustering in multiple dimensions on top of pricing.

Figure 2.4 shows the 1078 product listing unit price. Outliers can be easily filtered out.

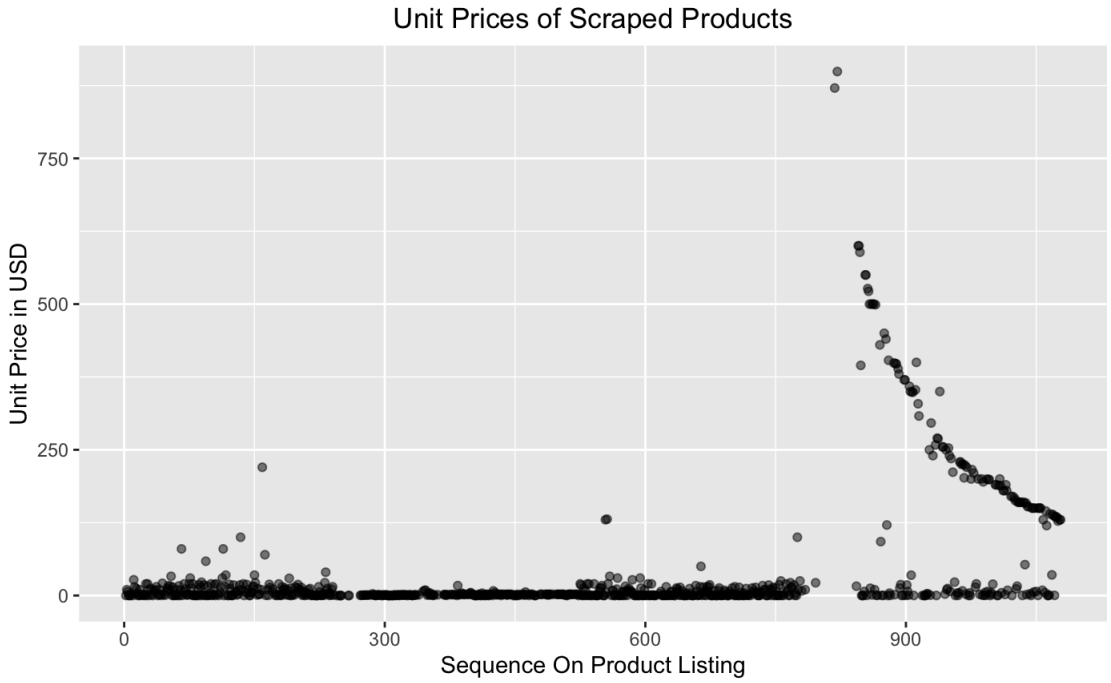

Figure 2.4: Face Mask Listings Unit Price Distribution

2.4.2 Cutter

One simple approach missing from current ASC research papers, which go to either a full sentence as in Weakly Supervised[4] or to the phrase or word level as in Snippext[38], is to cut text units into a minimal aspect-sentiment level, as to reduce chances when more than one aspect co-exist in one sequence and the Machine has to make a softmax hard call on what is the most likely aspect that this text unit is classified into. The execution is easy, as shown in Figure 2.5 and more details are provided in the algorithm section (chapter 3.3.2).

There are unfinished-topic-sequence-cut-too-early situations. In the sentence "The colors are bright and cute", "cute" was grouped with the segment after "and". Or there are still multiple aspects chain connected by "or". For the most part, this simple text cutter modulates text sequence to the right semantic units which aspect and sentiment Classifiers can better operate on.

A Review's Journey

Original Text

ASIN: B08HLHT8HG **Location:** 3rd review on page 31 **Verified Purchase:** Yes
Title: A welcome change **Date**: Nov 10, 2020 **Helpfulness Vote**: 1
Body: People have actually stopped me in the stores and stuff to ask where I got htese masks. The colors are bright and cute- it's a very welcome change from the blue masks everywhere. They are not too thick, but they still feel substantial enough to feel safe. The ear straps are comfortable. The nose piece bends easy around your nose to fit snugly. The price is very good for this amount of masks.

Collector-Cutter → Classifier → Compiler

rID: 193.31.3

Row	Text	Aspect		Sentiment	Aspect Count	Ratio
92058	People have actually stopped me in the stores	0	non-aspect	0 negative		
92059	stuff to ask where I got htese masks	0	non-aspect	1 neutral	56,539	14.41%
92060	The colors are bright	9	color	2 positive	98,47	2.51%
92061	cute- it's a very welcome change from the blue masks everywhere	8	look	2 positive	8,041	2.05%
92062	They are not too thick	2	thickness	2 positive	16,555	4.22%
92063	they still feel substantial enough to feel safe	12	protection	2 positive	7,841	2.00%
92064	The ear straps are comfortable	21	earloop	2 positive	11,202	2.85%
92065	The nose piece bends easy around your nose to fit snugly	23	fit @ nose	2 positive	5,514	1.41%
92066	The price is very good for this amount of masks	6	value	2 positive	14,188	3.62%
Total	**Review Text**				392,366	100%

Figure 2.5: One Review Example Through the SweetSpooter Pipeline

2.4.3 Classifier

After retraining several language models, for the dataset size and type of classification task, bert-base-uncase outperformed Snippext, bert-base-cased, or even bert-large-cased. More details will be discussed in the Methodology chapter.

For the classifier to perform well, the quality of the training dataset, especially when the dataset is small, proved to be crucial. Labeling with consistency, accuracy, clearly defined and non-overlapping categories, the awareness of the implicit and the explicit context of what Machine reads and what Human reads all contribute to the quality of the dataset, hence the performance of the trained classifier.

2.4.4 Compiler

As the classification of aspect and sentiment was trained and executed separately, the end result in which every single row is classified on two dimensions was compiled by attaching both aspect and sentiment classification results in the same row. With this result, Business Analysts can visualize and dig out valuable insights after classifying nearly 400,000 text units.

They also have the option to drill down to the original text under a particular category, such as negative reviews on "odor" which ranked the top on the negative rate.

2.5 Conceptualization

To describe the SweetSpotter pipeline process in simple math terms, the notations of each entities in the computation are defined here.

2.5.1 Collector

$R_{pq} \in N$ denotes each review entry in this corpus of reviews from N product listings with its review date, helpfulness, review title and review body. R_{pq} is the q^{th} review of n reviews in the the p^{th} product on the N listing. $p \in [1, N]$ and $q \in [1, n]$. $N = 536$ in this thesis. Product listing was filtered down to "face mask listing with reviews after de-duplication". $n \in [1, 5000]$, when the number of review count for each product p varies. R_p means all n reviews for product p.

2.5.2 Cutter

A review entry R_{pq} is further split into $r_i \in m$ where m is the number of review chunks R_{pq} is divided into. $m \in [1, 181]$. Our dataset is made up of the review text body of each r_i from R_{pq}. M is all m review text units from R_{pq} collected together: $M = 392366$.

2.5.3 Classifier

Each $r_i \in M$ is classified by Language Models of choice, most notably BERT. BERT (Bidirectional Encoder Representation for Transformers) is a neural network that allows each word to pay attention to all the other words in the sequence **simultaneously** instead of **sequentially** as in RNN (Recurrent Neural Network). The framework is based on the "attention" mechanism established by the transformer in the well-known paper "Attention is

All You Need"[50]. Such "attention" values are represented by attention weights. BERT innovated with its "bidirectional" attention assigning, in which each word gives and receives attention from both before and after the position where the word sits at, hence the name "bidirectional". Attention mechanism provides a computationally viable solution to contextualize for individual words so that each word has a one-on-one relationship with all the other words in that context. This mimics how humans process information. Though we hear speeches or read texts word by word, we make sense of meanings after all words are received and processed. Partial hearing or reading without the full context can often mislead us, even if humans are good at predicting the full meaning with incomplete information. Without getting into the details of how the BERT neural network architecture works with its neural network layers, attention weights, and the multiple "heads" in this thesis, highlighted here are the two most important concepts at work behind the computation of how BERT trains.

In **Forward Feeding**, BERT predicts the most likely class for an input text sequence. This thesis is a 31-class Aspect and 3-class Sentiment multinomial multivariate classification. **Softmax**, instead of **Sigmoid**, is used to compute the likelihood for each class.

Let us denote A as the Aspect class space. There are 31 classes, indexed from 0 to 30. Each Aspect class is denoted as a. S as the Sentiment class space includes negative, neutral, and positive, which are indexed as 0, 1, 2. Each Sentiment class is denoted as s.

The Aspect classification process is illustrated here as an example. For Sentiment classification, just switch out a to s and A to S.

$$P_a = \frac{e^{h_a}}{\sum_{a=1}^{A} e^{h_a}}$$

P_a is the probability of the input text sequence belonging to class a. h_a is the "a" class's last hidden layer value. The denominator is the summation of the exponentiated h_a for Softmax to scale the probability within 0 to 1, i.e. from least to most likely the input text belongs to a class.

To simplify the formula, the denominator $\sum_{a=1}^{A} e^{h_a}$ is set to equal Z. The original Softmax

equation can be re-written as

$$P_a = \frac{e^{h_a}}{Z}$$

In **Backward Feeding**, where **Loss Function** is used to find out the difference between the predicted value and the labeled "true" value so to adjust the parameters to make a better prediction in the next round of Forward Feeding, the prediction optimization is conceptualized to seek the **Maximum Likelihood** for all classifications multiplied together, as in below formula.

$$\prod_{a=1}^{A} P_a^{y_a}$$

In Maximum Likelihood, each Aspect class' probability P_a is set to the order of the actual label for that Aspect class $y_a \in \{0, 1\}$, a binary space. A Classifier's job is to maximize the likelihood of a correct prediction by the multiplication of all class probabilities to the order of 0 or 1. The result would be the probability of P_a for an a Aspect which actual label is $y_a = 1$ as the rest of the probabilities comes to $\prod_{i \neq a}^{A} P_i^{y_i} = \prod_{i \neq a}^{A} P_i^0 = 1$. The closer P_a approaches to value 1, the stronger or more confident the Classifier becomes.

To facilitate computation, multiplication is converted to summation by taking a logarithm, hence the **log-likelihood** formula:

$$\sum_{a=1}^{A} y_a \cdot log(P_a)$$

To **maximize** the above log-likelihood is equivalent to **minimize** the negative version of it, which happens to be **cross-entropy** equation:

$$-\sum_{a=1}^{A} y_a \cdot log(P_a)$$

With the earlier definition of $P_a = \frac{e^{h_a}}{Z}$, the log-likelihood can be re-termed to:

$$
\begin{aligned}
&= -\sum_{a=1}^{A} y_a \cdot log(\frac{e^{h_a}}{Z}) \\
&= -\sum_{a=1}^{A} y_a \cdot (h_a - log(Z)) \\
&= -\sum_{a=1}^{A} (y_a h_a - y_a log(Z)) \\
&= -(\sum_{a=1}^{A} y_a h_a - log(Z))
\end{aligned}
$$

Now to differentiate the above cross-entropy on one class by replacing a with k to make it clear that it is operated on one k class, it can be shown that the Maximum Likelihood Loss Function derivative ends up in the exact Error Loss function derivative for each class as in a Regression classification with Least Squared Loss Function.

$$
\begin{aligned}
\frac{\partial Loss}{\partial h_k} &= \frac{\partial[-y_k h_k + log(Z)]}{\partial h_k} \\
&= -y_k + \frac{1}{z} e^{h_k} \\
&= -(y_k - P_k)
\end{aligned}
$$

2.5.4 Compiler

All the classification results are collected into multinomial multivariate classes. To normalize the ratio of each class across the summation of classes, two ratio formulas were used. Denote Ω as the full set of Aspects in 31 classes and ω as the subset of Aspects in 29 classes, excluding "non-aspect" and "product" general classes. The reason for using ω subset is to compute a ratio of a specific feature class over the range of all specific features, instead of being diluted by almost 38% of non-specific classes.

The all-inclusive class ratio is all classes computed by normalizing over Ω classes.

$$
Ratio_a = \frac{Count(a)}{\sum_{a=1}^{\Omega} Count(a)}
$$

The more useful class ratio is feature-specific classes in ω classes.

$$Ratio_a = \frac{Count(a)}{\sum_{a=1}^{\omega} Count(a)}$$

Finally the performance metrics used in this thesis are formulated as below where t is the test or validate dataset size, simply all correctly classified divided by the total classified.

$$accuracy = \frac{\sum_{i=1}^{t} 1 \cdot (\hat{y}_i = y_i)}{t}$$

To get to F1, two metrics that make up F1 are explained first. Here a shorthand of tp as True Positive and tn as True Negative are correctly classified, while fp as False Positive and fn False Negative are wrongly classified.

$$Precision = \frac{tp}{tp + fp}$$
$$Recall = \frac{tp}{tp + fn}$$
$$F1 = 2\frac{Precision \cdot Recall}{Precision + Recall}$$

Weighted-F1 is when the Precision and Recall are weighted by multiplying the number of $tp + fn$ in each class. The other commonly used macro-F1 averages all classes Precision and Recall without weighing them per the class size. Weighted-F1 is used in this thesis.

CHAPTER 3

Methodology

3.1 Snippext

Snippext pipeline addresses the small-dataset-size issue by tagging aspect and opinion words or phrases and shuffling with replacement in one of the eight new data creation rules. There are two levels of data augmentation: with a labeled dataset, switch in linguistically similar words at the right placement within text units; with the unlabeled dataset, tag-identify such aspect or sentiment words and generate the new dataset. In paper[38], it achieved slightly better performance than the baseline model bert-base-uncased.

The result is not what this ASC project came to. On the contrary, the bert-base-uncased outperformed Snippext on three out of four tasks as shown on Table 3.1. In preparing the dataset, I noticed how mechanical this approach processes language. Its aspect-sentiment pairs are mostly in the *adjective + noun* structure. In the "face mask" dataset, the *adjective + noun* pairing is a rudimentary English expression, used mostly by non-native speakers. Native speakers who have a higher command of the language tend to express their opinions about a product attribute in diverse and often indirect ways.

12 types of English expressions are identified to be problematic to fit into the Snippext tagging framework. B-AS denotes Aspect at the Beginning of a sentence, B-OP Opinion at the Beginning of a sentence, I-AS Aspect in the middle of the sentence, I-OP Opinion in the middle of the sentence.

Type 1: B-AS in adjectives B-OP in verbs

Text: “These masks feel really comfortable”.

Tasks and Performance Snippext vs BERT

Task	**Metrics**	**Snippext**	**Bert**
tagging	P/R	**72.848/66.778**	70.843/69.111
pairing	A/P/F1	0.699/0.644/0.776	**0.740/0.727/0.771**
aspect	A/MF	0.709/0.615	**0.759/0.703**
sentiment	A/MF	0.880/0.838	**0.887/0.853**

Table 3.1: Snippext MixDA and Bert Base Uncased on Face Mask Dataset

Adjective "comfortable" is the Aspect "comfort". Verb "feel" is a confirmation that this statement comes from the user's personal experience. The noun "these masks" are is not the Aspect. When a more specific Aspect is present, the generic Aspect will be overlooked.

Type 2: B-AS in nouns B-OP in adjectives

Text: "Comfortable fit"

Noun "fit" is Aspect "fit", when comfortable can be Aspect "comfort" as well, but in this phrase, comfortable is "positive" Opinion. An alternative expression can be "the fit is comfortable." It is not necessarily the *adjective + noun* structure.

Type 3: B-AS in adjectives B-OP in adverbs

Text: "well made product"

Verb turned Adjective "made" is Aspect "workmanship", while Adverb prefix to the Adjective "well" is "positive" Opinion. "Product" as the generic Aspect is overlooked.

Type 4: B-AS set to "product" when purchase occasion is mentioned

Text: "added to the baby shower bag"

This is not relevant to language, but the use of the product. Purchase for a particular occasion is classified as "positive" on the "product" when no specific feature of the product is present in the text.

Type 5: B-AS and B-OP in disconnected sequence

Text: “fit more snuggly over mouth” (typo is kept here)

"fit ... over mouth" labeled as Aspect "fit @ mouth" is interrupted by Opinion "more snuggly". Snippext tagging only takes in a continuous text sequence and cannot leap over non-group words.

Type 6: B-AS set to "product" when only B-OP is present

Text: “they are over the moon”

In the context, “they” mean the user’s family, not masks. It’s sentiment positive without aspect words. It is labeled as “product" and "positive”.

Type 7: B-AS in adjectives and B-OP in nouns

Text: “ill-fitting garbage”

"fitting" is the Aspect "fit" and both "ill" and "garbage" are Opinion "negative". This is the reverse of the *adjective Opinion + noun Aspect* paring structure as Snippext maintains.

Type 8: B-AS itself is B-OP positive

Text: “pretty”

"Pretty" is B-AS "look" and B-OP "positive" two dimensions folded into one word.

Type 9: B-OP implied by syntax or semantics

Text: “to thick for me to be able to breathe” (typo is kept here)

Expressions such as "too. . . to. . . , had to. . . , would. . . , till. . . , thought. . . , wish. . . " implies often the reversed opinion. To human readers, "too thick ... to breathe" means "negative" on "breathability". To the Snippext tagging, it is mission impossible. Expressions in simple English words can be a negation of the literal meaning. And these words would be removed as "stop words" by the standard text mining procedure.

Type 10: When multi AS in a sequence, classify with B-AS.

Text: “kids didn’t complain about them being too itchy or uncomfortable or hot”

When multiple Aspects co-exist ("itchy" is aspect "softness"; "uncomfortable" aspect "comfort"; "hot" aspect "breathability"), classify this text unit as "softness", the first one

showing in this "or"-connected aspect chain. Unless per hierarchy, the more specific lower-level got classified, than the general upper level class. The Cutter singles out B-AS in most cases, but did not split on “or” connectors. So when multiple Aspects are present in one text unit, it is labeled per the first Aspect.

Type 11: Review context missing

Text: “making as few adjustments as possible”

Context: before this text segment, the reviewer talked about broken nose wire. The advice the reviewer gave is to make minimal adjustments to avoid nose wire break. "as few ... as possible" would have been labeled as Aspect "nose wire" and Opinion "negative" in context. Without the context, it is classified as "ease to adjust" Aspect and "positive" Opinion.

Type 12: Human life context

Text: “Mask kept falling down”

There is no Aspect word, but the whole sentence described an ill-fit situation. There is no way to tag any individual words in this sentence to tell the Machine it is "fit" and "negative" per the Snippext framework.

To make labeling less muddy to the human classifier, for anything classified as "non-aspect", the sentiment default is set to 0. This made a big difference when training the sentiment classifier. More details on dataset adjustment are discussed in the Results chapter.

Machines have no such human life context, as machines do not use “face masks” or live through a pandemic. All the societal, political, cultural, or biological context on top of the intricacy of human languages are lost on Machines. Humans by manually labeling a training dataset share such contextual knowledge with Machines. Snippext is too narrowly rule-based to capture the context in life or in language by measuring text on the yardstick of AS and OP. It misses out a big part of the meaning making process.

3.2 Huggingface

Huggingface provides a repository for all popular language models, pre-packaged for use. Just fine-tune with the dataset for a particular task and apply the trained model to the full dataset. Of its many language models, I picked bert-base-uncased, bert-base-cased and bert-large-cased for model comparison. From the training result, bert-base-uncased performed best. Accuracy and weighted F1 were used as performance metrics, because recall in the 31-class multinomial classification would not have been useful.

3.3 Algorithm

It took some exploration to find out the best way to execute the codes in both R and Python throughout the 4C SweetSpotter pipeline. Here is a high-level summary of the algorithms used to process review texts through the SweetSpotter pipeline.

3.3.1 Collector

There are two Collector steps: product listings and product reviews. The process is similar. Here the individual product review scraping process is illustrated.

Input: parsed data after getting HTML page of each R_p from Amazon

Output: csv file with product review attributes

Process:

for a product in all product listings

    connect to product p URL

    set up a container to collect R_p data

    for page in the total number of pages

        get review data, helpfulness, review title, review body on R_{pq}

save to the container each R_{pq} attributes

loop to the bottom of the page

append up to 10 R_{pq} on this page to R_p csv file

click to the "next page"

3.3.2 Cutter

After a few trials and errors, the best formula to cut paragraphs into sub-sentence level semantic units is below.

Input: scraped review body in R_{pq} from each product R_p file

Output: review text units r_i in each R_p file

Process:

for each R_p from N files

read in all R_{pq} from R_p file

set up an empty container for R_p to receive text units

for each R_{pq} in all R_p

set up a mini container to receive text units

split R_{pq} by separators including **and, but, &, . , !, ,** into r_i

save all r_i into the R_{pq} container vertically

keep all the other review attributes from R_{pq} on each r_i row

append r_i with all R_{pq} attributes to the R_p container

3.3.3 Classifier

There are two Classifiers: Aspect and Sentiment. The process is the same. Here illustrated is the Aspect Classifier algorithm.

Input: subset (training) or full (predicting) $r_i \in M$

Output: classified a label for each r_i

Process:

Training Step:

load in required packages (pandas, numpy, transformers, torch, sklearn)

load in a training dataset, as subset of $r_i \in M$

tokenize r_i texts

convert labels from verbal to numerical

create validate dataset with sklearn train_test_split

create performance metrics with sklearn and numpy

create train and validate datasets with torch

train with the Language Model of choice

set arg parameters such as the number of epochs, steps

save the best model

Predicting Step:

load in $r_i \in M$

tokenize texts

set all Y labels to "0"

create the X_dataset

load in trained model

predict with the trained model on the X_dataset

convert the predicted values to predicted labels with numpy

save the predicted labels to the $r_i \in M$ file

3.3.4 Compiler

Combine the classification results from the Classifiers and compute the class distributions. Below shows the starting point of the analysis of the results: getting ratios as a way to normalize and compare on the same basis how frequently each Aspect is reviewed and how positive or negative reviewers collectively talked about any Aspect.

Input: $r_i \in M$ with classified $a \in A$ and $s \in S$ where $A = 31$ and $S = 3$

Output: $Ratio_a \mid a \in \Omega$ and $Ratio_a \mid a \in \omega$ where $\Omega = 31$ and $\omega = 29$ and $Ratio_s \in S|a$

Process:

for all r_i classified

count a and s as shown in Figure 5.6 Count columns

calculate $Ratio_a$ two ways:

across all classes $a \in \Omega$ on the Aspect Axis

across specific features classes $a \in \omega$ on the Aspect Axis

calculate $Ratio_s$ within each a where all 3 s ratios added up to 100% on the Sentiment Axis

---

CHAPTER 4

Experiment

4.1 Data Scraping

In June 2021, search word "face mask" listings were scraped with the product name and url address. Because Amazon has a ceiling of 306 listings per search word display, all sorting options were scraped. Total 1419 "face mask" listings were collected. After removing duplicates per name and url address, a total of 1078 listings were the final step-1 scraping result.

In July 2021, crawlers were developed to collect two types of information from web pages on the 1078 listings:

- basic product information, including product name, pack size, selling pricing, posting date, manufacturing information, features as listed by manufacturers
- full-depth reviews per each listing from the first review to the last review on the last page. Each review entry included review title, review date, review helpfulness, purchase verified, and review body.

Of the 1078 listings, 935 are pandemic-use face masks and out of these, 536 contain text reviews. Review counts per product vary from 1 to 5000. The mean is 143 and the median 30. A total of 78,191 reviews on the 536 products were collected at the end of step two.

With the text Cutter, 392,366 text units were parsed out from the 78,191 reviews, separated by period, comma, and connector words like "and" and "but".

Text Entity Counts At Difference Stages

Text Types	**Counts**
product listings	1,491
unique product listings	1,078
face-mask product listings	935
face-mask listing with reviews	536
face-mask review documents	78,191
face-mask review text units	392,366

Table 4.1: Text Entities Size Change through Collector

4.2 Dataset Labeling

Training dataset was semi-randomly picked from the review text body, to choose the most "representative" reviews from the nearly 80,000 review entries. Product listings with a modest review count (<150), an average rating (close to 4.6), three unit price levels, and various positions on the scraping sequence, were picked to make the training dataset. Whenever multiple options were meeting the criteria, I would close my eyes and touch the screen. Whichever row of a product my finger landed on was “randomly” selected. If any class is under-represented (<7) in the training dataset, keyword search among the review texts would provide more text options related to that class and a supplementary selection would fill out each class to a minimum of 7 samples.

Multiple rounds of re-labeling improved the quality of the training dataset, in terms of accuracy, consistency, and product-engineering-based taxonomy of classes. The first round of relabeling combined labels into the granularity which product designers can use to find the sweet spots. The later relabeling further clarified the class scoping to stay as unambiguous, machine-friendly, and text based rather than context-based (from the rest of the review body) as possible. After careful curating, Aspects were finalized into 31 classes, as shown Figure 4.2 and 4.3.

Four major types of Aspect Groupings:

- **General labels**: irrelevant texts like "as advertised" as "non-aspect" and non-feature-specific texts as "product"
- **Product Engineering fundamentals**: material, construction, look, each of the three with its subcategories
- **Manufacturing Performance metrics**: attributes related to the manufacturing process, including value, workmanship, packaging, quality
- **User Experience**: composite user experience factored from product engineering or manufacturing cannot be singled out into any of the above Groupings. For example, "comfortable" can be a combination of good fit and soft and breathable material.

4.3 Colab Computing

After the training and testing datasets were consistently and correctly labeled after several cycles of one-person self auditing, a pipeline of fine-tuning and predicting with the Huggingface transformer language models was developed to take in the training dataset and evaluate on the test dataset, save the best performing model, use that model to predict the remaining of the nearly 400,000 text units. Colab free tier is enough to compute this 22M dataset. It could take days without GPU, but hours with GPU on the resource.

Aspect Classification Scoping

Aspect	**Text Examples**
General	
non-aspect	non-product, context, social commentary, foreign language
product	general (dis)liking, use occasion, purchase/return
Material	
thickness	filter, layers, insert
breathability	hot, breathe, sweat, acne
material	fabric, liner
softness	itchy, scratchy, irritating, tickling
lightweight	thin, light
Construction	
fit	run small/big, small/big on someone, head
fit @ face	chin, sides, face
fit @ ear	pull, hurt ear, too long/short
fit @ mouth	cover, press/touch/room to mouth, chin
fit @ nose	snug, seal, hold
fogging	fog up, can see
Look	
look	go with outfit, cute, receive compliments
color	specific color
embellishment	print, sequins, lace

Table 4.2: Example Texts Classification Scoping Part I

Aspect Classification Scoping

Aspect	**Text Examples**
Manufacturing	
value	price, worth, deal, expensive
quality	general quality
workmanship	made, ripped, came off
packaging	shipped as ordered quantity, color, box condition
earloop	broken, missing, made, rip, tied
nose wire	missing, broken, showing, flimsy, rigid
User Experience	
comfort	due to fit, fabric, breathability
protection	safe, protect, medical/surgical
odor	smell
sturdiness	ripped, durable
sterility	clean, sanitary, filth
ease to care	wash, dry
ease to carry	individually packed, store
ease to wear	put on
ease to adjust	adjust, knot, cut

Table 4.3: Example Texts Classification Scoping Part II

CHAPTER 5

Results

5.1 Aspect Classification

Through 25 epochs, an 83% accuracy and close to 0 loss was achieved on the bert-base-uncased model.

Figure 5.1: Aspect Classifier Training Metrics By Epochs

Table 5.1 shows the performance comparison among 3 language models. Model bert-base-uncased outperformed the other two models, with a relatively smaller parameter and model size.

Confusion Matrix from bert-base-uncased is shown in Figure 5.2. Even if it was a 31-class classification, BERT correctly classified most test dataset. "ease to adjust" (indexed

Aspect Classifier Performance Metrics

Parameter	**bert-base-uncased**	**bert-base-cased**	**bert-large-cased**
Dataset Size	1431	1431	1431
Epoch	25	25	25
Training Loss	0.0036	0.0026	0.0008
Validation Loss	1.08583	1.19651	1.455382
Accuracy	**0.829268**	0.773519	0.804878
Weighted F1	**0.82705**	0.771738	0.804489
model.bin size	438.1M	433.4M	**1.33G**

Table 5.1: Performance Metrics of Three BERT Models on Aspect Classifier

16) was confused with "product", "fit" or "earloop" and "fit @ nose" (indexed 23) was wrongly labeled to "nose wire" and "ease to adjust". This could be from the text itself, it is more difficult to tell the different perspectives when classifying highly overlapping words into different classes. "ease to adjust" emphasizes the motion "adjust", when "nose wire" and "earloop" are about the material and construction of the piece on a mask.

5.2 Sentiment Classification

As this is a 3-class classification task, the starting loss was already within 1 and approached 0 after 750 steps.The accuracy and weight F1 both stabilized around 0.90 after 800 steps, after a 0.91 uptick at step 500.

Three versions of the training dataset were used to improve Classifier performance.

- At 1049, it was the original training dataset and achieved < 0.85 accuracies.
- At 1431, the test dataset was combined into the training dataset before splitting into train and validate dataset. This 36% size increase didn't see higher accuracy on bert-base-uncased.

A E	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
0	34	4	0	1	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	3	46	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	10	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	2	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	1	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	1	0	0	0	0	0	0	9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	1	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
17	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	1	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	4	0	0	0	0	0	0	0	0	0	0	0
20	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	1	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	4	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	3	0	0	0	0	0	0
25	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
29	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 5.2: Aspect Classifier Confusion Matrix

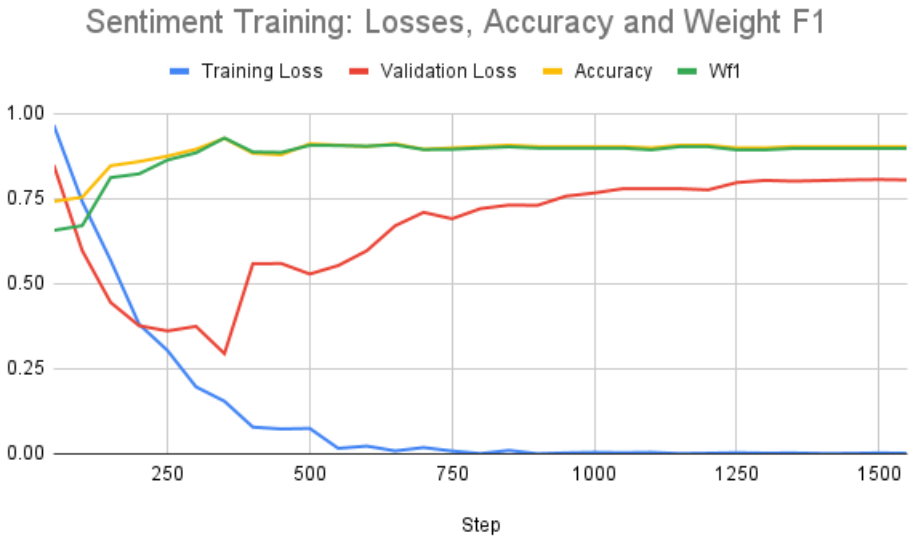

Figure 5.3: Sentiment Classifier Training Metrics By Epochs

- At 1236, when 195 (almost 14%) "non-aspect" was removed from the training dataset, the performance jumped to 0.91 on both bert-base-uncased and bert-large-cased. All "non-aspect" text units uniformly labeled as "neutral" confused the Classifiers in earlier datasets because this was an arbitrary rule by the author instead of a text-based labeling.

Sentiment Classifier Performance Metrics

Parameter	**bert-base-uncased**	**bert-base-cased**	**bert-large-cased**
Dataset Size	1236	1049	1431
Epoch	25	25	25
Training Loss	0.0747	0.0045	0.0039
Validation Loss	0.528419	1.257125	1.405293
Accuracy	**0.91129**	0.847619	0.821678
Weighted F1	**0.906784**	0.840916	0.861094
model.bin size	438M	433.3M	**1.33G**

Table 5.2: Performance Metrics of Three BERT Models on Sentiment Classifier

The confusion matrix on shows that neutral (labeled as 1) is the most mis-classified. Negative is at 0.84 and positive at 0.96 accuracy rate.

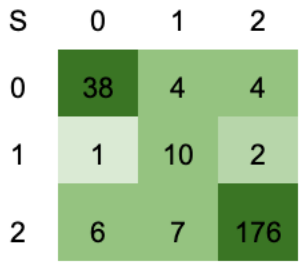

Figure 5.4: Sentiment Classifier Confusion Matrix

5.3 Compiled Results

With the best performing aspect and sentiment Classifiers, the nearly 400,000 text units were classified into one of the 31 aspect classes and one of the 3 sentiments. The below graph shows "non-aspect" which does not contain any general or specific product attribute defined earlier in the data labeling takes 14.41% of all text units, and general comments on the product is 23.49%. The useful consumer insights on a feature pre-defined in the product-engineering-based taxonomy takes only 62.10% of all review text units.

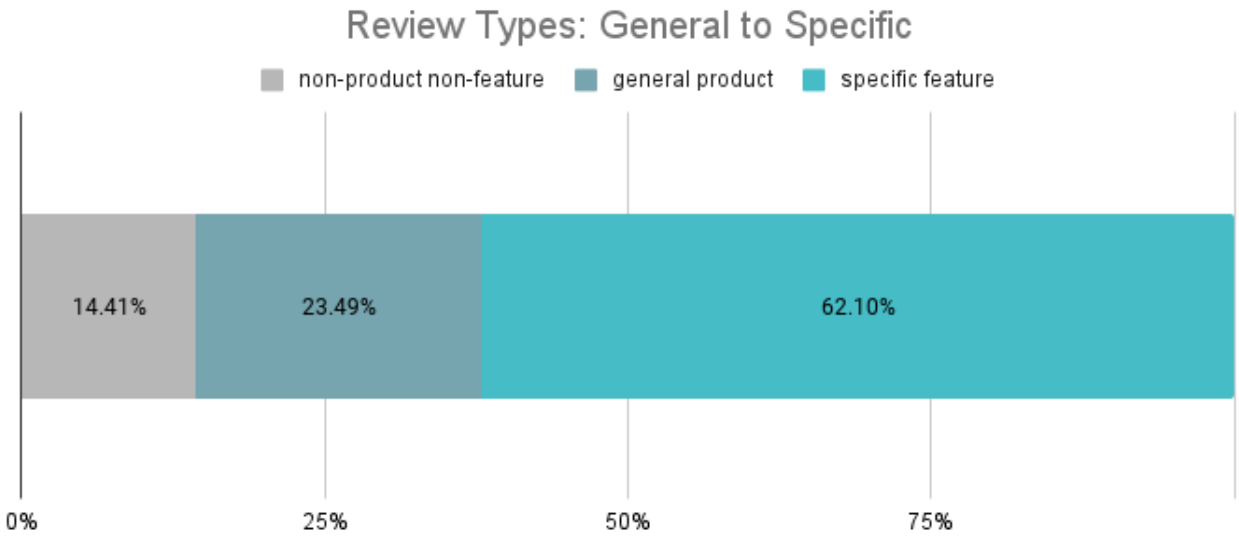

Figure 5.5: Review Groupings From General To Specific

Below are "non-aspect" text examples. Some ranted on pandemic events in public health policy or the global supply chain. Others established context before addressing a particular product feature. "product" text examples shows a general sentiment toward the mask, without mentioning any specific product features.

Figure 5.6 is the "big picture" of how many review texts were classified into each of the aspects and sentiments and what the aspect ratio in all 31 classes and the sentiment ratio among the 3 classes. From this color scaled classified chart of all classes, critical attributes can be easily seen: fit is the most talked about product attribute, above any location-specific

non-aspect
I specifically ordered them because I wanted ones on the CDC approved list
they haven't rested for one day before they changed their role to supply the world with China PPE supplies
I have ordered masks from different website since all of sudden the masks is so demanded
At 60 years old I have tried SO MANY products
product
I thought I would give this Zombie face mask a shot
–Have a feeling we'll be using these for a while this year so will try a different brand on the next order
I was so afraid to buy from someone with no feedback
they're going back
have been using them for a couple of weeks when I go out

Table 5.3: Review Examples Classified as "non-aspect" or "product"

fit classes. Odor is the least positive of all, but only 1.53% of all review texts talked about it. In this way, product engineers can have a numerical sense of how much a product attribute was valued, loved, or hated by consumers.

5.4 Business Intelligence

In this chapter, five data presentation and visualization examples demonstrated how to use SweetSpotter deliverables to inform product optimization. Such information can provide a powerful guideline for designers and manufacturers of face masks to optimize their products for the next cycle.

For example, 63% of the reviews talked about ear-loops negatively. Manufacturers can retrieve a desired number of reviews on this topic and see what exactly people said about it,

	Percentage				Count			
aspects	pct	negative	neutral	positive	total	negative	neutral	positive
product	23.49%	21.43%	2.99%	75.58%	92154	19745	2757	69652
non-aspect	14.41%	35.39%	14.85%	49.76%	56539	20011	8397	28131
fit	8.13%	38.67%	1.83%	59.51%	31881	12328	582	18971
comfort	5.27%	10.34%	0.12%	89.53%	20688	2140	25	18523
thickness	4.22%	43.12%	7.62%	49.26%	16555	7139	1261	8155
value	3.62%	29.40%	0.44%	70.16%	14188	4171	62	9955
breathability	3.31%	28.51%	0.51%	70.98%	12973	3699	66	9208
fit @ face	3.09%	46.29%	2.00%	51.71%	12126	5613	243	6270
earloop	2.85%	63.60%	2.96%	33.44%	11202	7124	332	3746
softness	2.63%	38.03%	0.38%	61.60%	10306	3919	39	6348
color	2.51%	14.22%	10.77%	75.01%	9847	1400	1061	7386
fit @ ear	2.27%	48.37%	2.19%	49.44%	8888	4299	195	4394
quality	2.15%	22.42%	0.17%	77.41%	8433	1891	14	6528
look	2.05%	8.84%	2.64%	88.52%	8041	711	212	7118
protection	2.00%	25.02%	1.05%	73.93%	7841	1962	82	5797
ease to adjust	1.58%	35.49%	4.35%	60.16%	6180	2193	269	3718
odor	1.53%	78.96%	1.26%	19.77%	6018	4752	76	1190
material	1.45%	37.36%	14.22%	48.43%	5683	2123	808	2752
fit @ nose	1.41%	39.92%	1.72%	58.36%	5514	2201	95	3218
lightweight	1.40%	8.52%	0.31%	91.17%	5506	469	17	5020
ease to care	1.34%	30.29%	10.64%	59.07%	5246	1589	558	3099
packaging	1.31%	50.87%	18.00%	31.13%	5123	2606	922	1595
nose wire	1.17%	50.16%	5.67%	44.17%	4605	2310	261	2034
workmanship	1.17%	36.68%	0.98%	62.34%	4575	1678	45	2852
fogging	1.01%	36.03%	1.41%	62.56%	3969	1430	56	2483
ease to carry	0.97%	14.65%	3.13%	82.22%	3796	556	119	3121
fit @ mouth	0.85%	41.82%	3.16%	55.01%	3350	1401	106	1843
embellishment	0.84%	23.78%	10.11%	66.11%	3293	783	333	2177
sturdiness	0.82%	29.96%	0.19%	69.85%	3224	966	6	2252
ease to wear	0.61%	29.07%	10.13%	60.80%	2408	700	244	1464
sterility	0.56%	41.51%	0.81%	57.68%	2214	919	18	1277

Figure 5.6: Assembled Classifications after Being Compiled

in its original text, from the SweetSpotter classified database.

5.4.1 Grouped Aspects

Figure 5.7 visualized all feature-specific aspects (excluding "non-aspect" or "product") grouped by color and ordered by ratio scale to give product engineers a holistic view about all things talked about on "face mask" in pre-defined categories.

Remember the three fundamentals of Industrial Design? Fit, look, and material. On the pandemic-use "face mask", the valuation of the three major product attributes are ordered and weighted exactly as fit 26.97%, material 20.94%, and look 8.69%. This gives a strong signal for product engineers to improve on the fit of face masks on diverse human face sizes and shapes beyond just adults and children sizing. For wearable CPG like shoes or dresses or rings, there are well standardized sizing to cover somewhere near two standard deviations of the human body types and sizes, beyond which consumers have to tailor-make to fit a particular body type. It is no surprise that when mask wearing becomes universal during a pandemic, the former small market use (mostly in the medical or food preparation industries) wearable products all of a sudden need to fit all kinds of human faces. Maybe they won't get fine-grained as shoes, but the current one-size-fits-all problem is a top issue for consumers. Material is also weighted high because masks are worn on faces that are more tactile and olfactory sensitive than other parts of the body. That explains why softness, breathability, and odor-free are highlighted issues for consumers. Manufacturing grouping shows "value" is an appreciated feature because for one-time use masks the long-term aggregated cost can be significant. The least mentioned is about look, below 9% of all reviews.

In a pandemic, the purpose of a "face mask" is to provide protection (correlating to fit and thickness) comfortably (breathability and softness) at a good value. The usual fashion industry appearance-centric product design does not transfer well to "face mask". Some sellers are up-selling with more look-based design elements, such as embroidery or print or kids-friendly designs. Benchmarking against this collective consumer voice, it is loud and clear that look is not that important for a face mask during a pandemic.

5.4.2 Top Negative

On a top negative chart (Figure 5.8), a business can find what are the most disliked product attributes. "Odor" far exceeded others in its negativity. It makes sense for people to be odor-sensitive because face masks are right over their noses. Especially for the value one-time use face masks, how to address the odor issue can make a big difference. "Earloop" came up next. Many complained about how ear loops were either missing or broken off, which immediately made the face mask unwearable. "Packaging" is more of a logistic challenge in that people often received orders different from what they ordered, either by quantity or by color. This top negative list can help designers and manufacturers allocate their resources to solve the worst problems per collective consumer experience.

5.4.3 Sorted Classifications

From the aspect ratio in Figure 5.9, one can tell how much "weight" consumers collectively put on a product feature by spending time and space commenting on it. It is a very powerful tool in reading through a large corpus of consumer reviews to get quantifiable and verifiable consumption attributes evaluation, in comparison to the consumer panel surveys in a laboratory environment. At such a scale, the scraped consumer reviews are still a sampling of the total user experience, but they approximate the "truth" in the population with much higher accuracy, simply based on extraordinarily large sample size.

5.4.4 Top Frequency Comments

This is a traditional text mining technique by purely counting the occurrence of the same wording. No bigram or trigram was applied here. This thesis only converted all words in lower case to show most commonly used expressions in a review unit. Top Frequency Comments (Figure 5.10) are one of the tale-teller signs in Fake Review Detection algorithms because fake review makers tend to repeat themselves and lack authentic expressions. So we take the list on its face value as the most commonly worded expressions used reviewing

"face mask".

5.4.5 Aspect "Odor" Sentiment Negative Reviews

The program can drill down by descending helpfulness the reviews that are classified as "Odor" and "Negative" to allow businesses to see in one place what consumers commented about in a selected product feature in a selected sentiment, as shown on Table 5.4.

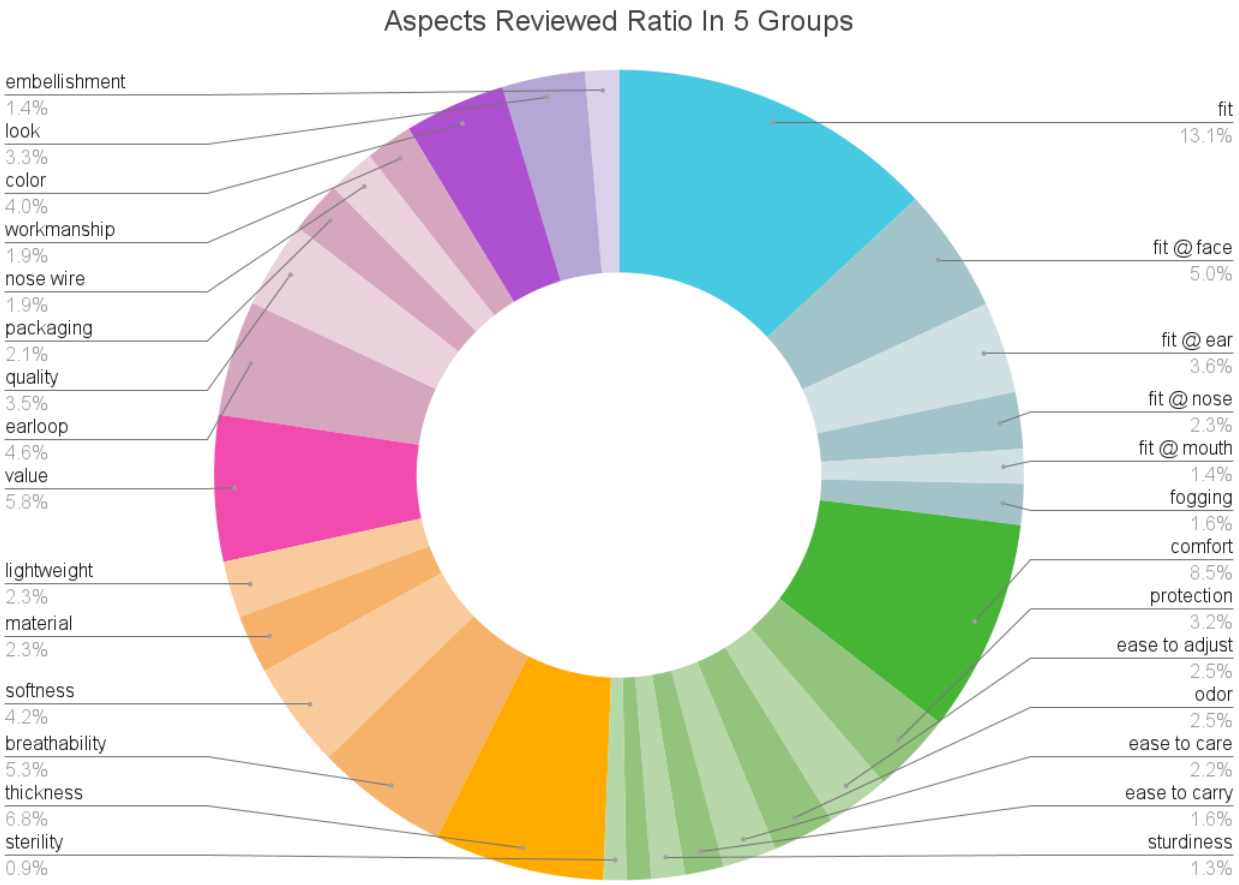

Figure 5.7: Grouped Classifications "Color Wheel"

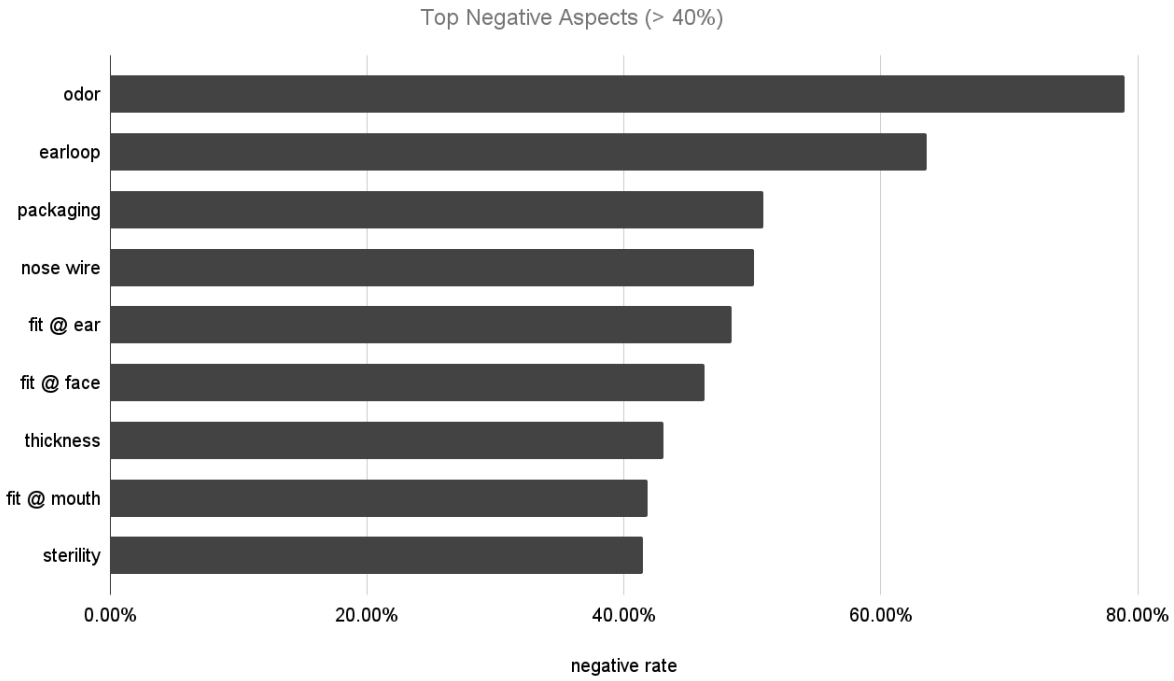

Figure 5.8: Top Negative Classifications

aspect group	group ratio	aspect	aspect ratio
construction	26.97%	fit	13.08%
		fit @ face	4.98%
		fit @ ear	3.65%
		fit @ nose	2.26%
		fit @ mouth	1.37%
		fogging	1.63%
UX	23.64%	comfort	8.49%
		protection	3.22%
		ease to adjust	2.54%
		odor	2.47%
		ease to care	2.15%
		ease to carry	1.56%
		sturdiness	1.32%
		ease to wear	0.99%
		sterility	0.91%
material	20.94%	thickness	6.79%
		breathability	5.32%
		softness	4.23%
		material	2.33%
		lightweight	2.26%
manufacturing	19.75%	value	5.82%
		earloop	4.60%
		quality	3.46%
		packaging	2.10%
		nose wire	1.89%
		workmanship	1.88%
look	8.69%	color	4.04%
		look	3.30%
		embellishment	1.35%
all aspects	100.00%		100.00%

Figure 5.9: Sorted Classifications

Review Text Units	Frequency
very comfortable	2191
good quality	960
light weight	910
great product	794
they are comfortable	723
fits well	579
great quality	556
well made	554
thank you	523
great fit	472
good product	427
great mask	425
great masks	411
fit well	399
comfortable to wear	397
good fit	387
great value	375
great price	352
highly recommend	351
love it	342
very soft	339
very breathable	320
love them	290
these are great	287
good price	285
they fit well	281
easy to wear	280
good value	273
love these masks	260
easy to use	257
they are very comfortable	239
waste of money	239

Figure 5.10: Top Frequent Text Units

Negative "Odor" Review Examples

ID	Helpfulness	Review Text
167.1.5	382	I didn't find the zombie mask to have an offensive smell
124.1.1	266	The mask has a bit of a plastic smell when first removed from its plastic bag
23.264.4	155	It has the same smell as the masks I got from my Dr
23.264.4	155	the feel of the fabric along with the smell of the mask is identical
167.2.6	144	the smell will go away as the mask dries
167.2.6	144	You will notice these masks don't have a very desirable smell to them
592.1.8	99	they do have an odor
212.1.1	91	When first opening the package they also had a very bad smell that took 2 washings to get rid of
166.2.1	77	The only negative I see is that these smell a little
978.1.7	47	The masks had strange smell that was annoying
978.1.7	47	If you are sensitive to smells
619.6.8	47	Little odor
50.1.5	44	The horrible smell WILL make you fall over
40.1.3	43	they smell terrible unless you vigorously hand wash them before use

Table 5.4: Review Examples Classified as "odor" and "negative" sorted by Helpfulness

CHAPTER 6

Summary

This thesis demonstrated how the SweetSpotter pipeline can quantify product features for product optimization with a pilot study on classifying the now universal experience of ordering and using face masks well documented in the Amazon consumer reviews. Business Intelligence informed by collective consumer insights can be used widely in any Consumer Package Goods sold on any e-commerce website with consumer reviews. The innovations of the SweetSpotter 4C (collector-cutter-classifier-compiler) pipeline in terms of minimal semantic unit cutting, product engineering and manufacturing informed labeling, and the business user-friendly deliverables not only made this Aspect-Sentiment-Classification task feasible in business applications but also achieved state-of-the-art classifier benchmark performance in 2021.

6.1 Limitations

Limitations to this research are bound by the current state of technology, understanding of the power and application of AI and ML, how the Machine world and human world parallel and interact, how reviews can be a biased sampling of broader user experience unwritten, unpublished, or uncollectible into this database, how fake reviews and the "natural selection" on e-commerce websites can skew reviews toward positive, where 4.6 is a 3, after rescaling.

Within the limited time and technology that the author commanded in 2021, the 4C pipeline can be streamlined and optimized to better garner collective human knowledge on a Consumer Packaged Goods domain.

6.2 Future Works

A lot can be done to expand the use of the SweetSpotter, such as including consumer uploaded product images, comparing manufacturer proposed benefits against consumer experience feedback, automating product classification to sort out different product categories under the same search phrase, automating de-duplication after scraping from different sorting orders, taking snapshots of consumer reviews at a certain frequency and detecting longitudinal trends, inserting pricing as a business dimension into the mix of other product attributes, and utilizing abstractive summarization to create a summary from the classified reviews.

6.3 Takeaways

This thesis, as an application of the current SOTA ML-powered NLP method in a real-world business use case to surface critical product attributes of "face masks" from the vast user experience captured in product reviews with its biases and limitations, demonstrated the importance to have real-world operational knowledge in a business domain (such as CPG product engineering) to construct a useful classification hierarchy which follows the Industrial Design principles and practices. I spent half a month on labeling, relabeling, reconciling, reducing, and finalizing into 31 categories for Aspect Classification. For any other product, a similar human expert labeling process is what makes the deliverables useful. This human involvement in the Machine Learning computation is the "quality guarantee" (to borrow a term from the CPG industry) for quality-in quality-out data processing. Especially when it comes to CPG product engineering, human experts provide the human-product-interaction contexts that Machines simply do not learn in a generic pre-training.

This thesis also demonstrated how powerful and accurate the SOTA ML method on the most pared-down version of BERT bert-base-uncased can be with a small one-man-shop scale dataset. BERT which made leaps in the NLP field with its bidirectional attention contextualization can pick up meaning cues embedded in human languages (such as *too... to...* expressions) more accurately than the over-engineered language processing commonly used

in traditional text mining techniques like lowering all cases, cleaning out typos or weird symbols, stripping texts of its meaning connectors and sense makers in “small words”. This is an exciting era for **human** + **machine** collaboration.

The goal of the SweetSpotter is to close that feedback loop by gathering consumer reviews and packaging them in an insightful and quantifiable way for designers, engineers and manufacturers to improve their next round of conception. "Face mask" is a relatively simple CPG product, which happened to be widely used during the pandemic and therefore quickly accumulated a large volume of review texts. To effectively sort through this huge body of reviews and classify them into organized information is what I hope the author achieved in this paper.

Bibliography

[1] Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. Unsupervised opinion summarization with content planning. *arXiv preprint arXiv:2012.07808*, 2020.

[2] Reinald Kim Amplayo and Mirella Lapata. Unsupervised opinion summarization with noising and denoising. *arXiv preprint arXiv:2004.10150*, 2020.

[3] Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata. Extractive opinion summarization in quantized transformer spaces. *Transactions of the Association for Computational Linguistics*, 9:277–293, 2021.

[4] Stefanos Angelidis and Mirella Lapata. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. *arXiv preprint arXiv:1808.08858*, 2018.

[5] Georgios Askalidis and Edward C Malthouse. The value of online customer reviews. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 155–158, 2016.

[6] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*, 2020.

[7] Roy Bar-Haim, Lilach Eden, Yoav Kantor, Roni Friedman, and Noam Slonim. Every bite is an experience: Key point analysis of business reviews. *arXiv preprint arXiv:2106.06758*, 2021.

[8] Arthur Bražinskas, Mirella Lapata, and Ivan Titov. Unsupervised opinion summarization as copycat-review generation. *arXiv preprint arXiv:1911.02247*, 2019.

[9] Arthur Bražinskas, Mirella Lapata, and Ivan Titov. Few-shot learning for opinion summarization. *arXiv preprint arXiv:2004.14884*, 2020.

[10] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[11] Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[12] Li-Chen Cheng, Kuanchin Chen, Ming-Chu Lee, and Kua-Mai Li. User-defined swot analysis–a change mining perspective on user-generated content. *Information Processing & Management*, 58(5):102613, 2021.

[13] Eric Chu and Peter Liu. Meansum: a neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning*, pages 1223–1232. PMLR, 2019.

[14] André Cibils, Claudiu Musat, Andreea Hossman, and Michael Baeriswyl. Diverse beam search for increased novelty in abstractive summarization. *arXiv preprint arXiv:1802.01457*, 2018.

[15] Maximin Coavoux, Hady Elsahar, and Matthias Gallé. Unsupervised aspect-based multi-document abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 42–47, 2019.

[16] Hoa Trang Dang. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12, 2005.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[18] Ill Chul Doo, Hyun Duck Shin, and Mee Hwa Park. Automated product review collection and opinion analysis methods for efficient business analysis. *International Journal of Computing and Digital Systems*, 10(1):37–45, 2021.

[19] Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409, 2021.

[20] Iftah Gamzu, Hila Gonen, Gilad Kutiel, Ran Levy, and Eugene Agichtein. Identifying helpful sentences in product reviews. *arXiv preprint arXiv:2104.09792*, 2021.

[21] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397, 2017.

[22] Sherry He, Brett Hollenbeck, and Davide Proserpio. The market for fake reviews. *Available at SSRN 3664992*, 2021.

[23] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28:1693–1701, 2015.

[24] Sharon Hirsch, Slava Novgorodov, Ido Guy, and Alexander Nus. Generating tips from product reviews. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 310–318, 2021.

[25] Brett Hollenbeck, Sridhar Moorthy, and Davide Proserpio. Advertising strategy in the presence of reviews: An empirical analysis. *Marketing Science*, 38(5):793–811, 2019.

[26] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, 2004.

[27] Hayate Iso, Xiaolan Wang, Yoshihiko Suhara, Stefanos Angelidis, and Wang-Chiew Tan. Convex aggregation for opinion summarization. *arXiv preprint arXiv:2104.01371*, 2021.

[28] Masaru Isonuma, Junichiro Mori, Danushka Bollegala, and Ichiro Sakata. Unsupervised abstractive opinion summarization by generating sentences with tree-structured topic guidance. *arXiv preprint arXiv:2106.08007*, 2021.

[29] Shreehar Joshi and Eman Abdelfattah. Multi-class text classification using machine learning models for online drug reviews. In *2021 IEEE World AI IoT Congress (AIIoT)*, pages 0262–0267. IEEE, 2021.

[30] Hitesh Kansal and Durga Toshniwal. Aspect based summarization of context dependent opinion words. *Procedia Computer Science*, 35:166–175, 2014.

[31] Nitish Shirish Keskar Bryan McCann Caiming Xiong Kryściński, Wojciech and Richard Socher. Neural text summarization: A critical evaluation. *arXiv preprint arXiv:1908.08960*, 2019.

[32] Alon Lavie and Abhaya Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the second workshop on statistical machine translation*, pages 228–231, 2007.

[33] Kevin Lerman, Sasha Blair-Goldensohn, and Ryan McDonald. Sentiment summarization: evaluating and learning user preferences. In *Proceedings of the 12th conference of the European chapter of the ACL (EACL 2009)*, pages 514–522, 2009.

[34] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[35] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[36] Alhassan Mabrouk, Rebeca P Díaz Redondo, and Mohammed Kayed. Seopinion: Summarization and exploration of opinion from e-commerce websites. *Sensors*, 21(2):636, 2021.

[37] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*, 2020.

[38] Zhengjie Miao, Yuliang Li, Xiaolan Wang, and Wang-Chiew Tan. Snippext: Semi-supervised opinion mining with augmented data. In *Proceedings of The Web Conference 2020*, pages 617–628, 2020.

[39] Guy Berger Craig Boutilier Morgan Currie Finale Doshi-Velez Gillian Hadfield Michael C. Horowitz Charles Isbell Hiroaki Kitano Karen Levy Terah Lyons Melanie Mitchell Julie Shah Steven Sloman Shannon Vallor Michael L. Littman, Ifeoma Ajunwa and Toby Walsh. Gathering strength, gathering storms: The one hundred year study on artificial intelligence (ai100) 2021 study panel report. http://ai100.stanford.edu/2021-report. Accessed: 2021-09-16.

[40] Jun-Ping Ng and Viktoria Abrecht. Better summarization evaluation with word embeddings for rouge. *arXiv preprint arXiv:1508.06034*, 2015.

[41] Nadav Oved and Ran Levy. Pass: Perturb-and-select summarizer for product reviews. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 351–365, 2021.

[42] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[43] Fatemeh Pourgholamali, Mohsen Kahani, Zeinab Noorian, and Ebrahim Bagheri. Learning product representations for generating reviews for cold products. *Knowledge-Based Systems*, 228:107282, 2021.

[44] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[45] Richa Sharma and Prachi Sharma. A survey on extractive text summarization. *International Journal of Advanced Research in Computer Science and Software Engineering*, 6(4):461–465, 2016.

[46] Yoshihiko Suhara, Xiaolan Wang, Stefanos Angelidis, and Wang-Chiew Tan. Opiniondigest: A simple framework for opinion summarization. *arXiv preprint arXiv:2005.01901*, 2020.

[47] Jiaxing Tan, Alexander Kotov, Rojiar Pir Mohammadiani, and Yumei Huo. Sentence retrieval with sentiment-specific topical anchoring for review summarization. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2323–2326, 2017.

[48] Hiren Kumar Thakkar, Prasan Kumar Sahoo, and Pranab Mohanty. Dofm: Domain feature miner for robust extractive summarization. *Information Processing & Management*, 58(3):102474, 2021.

[49] Artem Timoshenko and John R Hauser. Identifying customer needs from user-generated content. *Marketing Science*, 38(1):1–20, 2019.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[51] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016.

[52] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[53] Sobia Wassan, Xi Chen, Tian Shen, Muhammad Waqar, and NZ Jhanjhi. Amazon

product sentiment analysis using machine learning techniques. *Revista Argentina de Clínica Psicológica*, 30(1):695, 2021.

[54] FANGYU WU, ZHENJIE SHI, ZHAOWEI DONG, CHAOYI PANG, and BAILING ZHANG. Sentiment analysis of online product reviews based on senbert-cnn. In *2020 International Conference on Machine Learning and Cybernetics (ICMLC)*, pages 229–234. IEEE, 2020.

[55] Haibing Wu, Yiwei Gu, Shangdi Sun, and Xiaodong Gu. Aspect-based opinion summarization with convolutional neural networks. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3157–3163. IEEE, 2016.

[56] Wenting Xiong and Diane Litman. Empirical analysis of exploiting review helpfulness for extractive summarization of online reviews. In *Proceedings of coling 2014, the 25th international conference on computational linguistics: Technical papers*, pages 1985–1995, 2014.

[57] Euguene Yan. Bootstrapping labels via supervision human-in-the-loop. https://eugeneyan.com/writing/bootstrapping-data-labels. Accessed: 2021-09-30.

[58] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.

[59] Min Zhang, Brandon Fan, Ning Zhang, Wenjun Wang, and Weiguo Fan. Mining product innovation ideas from online reviews. *Information Processing & Management*, 58(1):102389, 2021.

[60] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.