Deriving Customer Experience Implicitly from Social Media

Aditya Kumar
aditya.kmr@flipkart.com
Flipkart

Sneh Gupta
sneh.gupta@flipkart.com
Flipkart

Ankit Sahu
sahu.ankit@flipkart.com
Flipkart

Mayank Kant
mayank.kant@flipkart.com
Flipkart

ABSTRACT

Organizations that focus on maximizing satisfaction, a consistent and seamless experience throughout the entire customer journey are the ones who dominate the market. Net Promoter Score (NPS) is a widely accepted metric to measure the customer experience, and the most common way to calculate it to date is by conducting a survey. But this comes with a bottleneck. The whole process can be costly, low-sample, responder-biased, and issues could be limited to the questionnaire used for the survey. We have devised a mechanism to approximate it implicitly from the mentions extracted from the four major social media platforms - Twitter, Facebook, Instagram, and YouTube. Our Data Cleaning pipeline discards the viral and promotional content (from brands, sellers, marketplaces, or public figures), and the Machine Learning pipeline captures the different customer journey nodes specific to e-commerce (like discovery, delivery, pricing) with their appropriate sentiment. Since the framework is generic and relies only on publicly available social media data, any organization can estimate its NPS after making suitable adjustments depending on the industry and geography. Our NPS model has a Mean absolute percentage error (MAPE) of **1.9%**, Pearson correlation of **79%**, and enables us to understand the actual drivers at the weekly level.

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**; • **Information systems** → *Sentiment analysis.*

KEYWORDS

Customer Experience, Net Promoter Score (NPS), E-Commerce, Social Media, Aspect-based sentiment analysis (ABSA), Transformer-based model - T5, Natural Language Processing (NLP), XGBoost, Ensembles, Machine Learning (ML), Information Extraction (IE)

1 INTRODUCTION

A superior experience results in customer loyalty, a long association with the brand, and positive word of mouth, which could be the key differentiators in building long-term relationships and continued success with the customers. The customer journey begins with brand awareness and goes through the cycle of engagement on the platform through Catalog discovery, Selection, Pricing, Delivery, Installation, and other nodes (depending on the industry). Knowing the customer experience across different nodes could help a lot in building appropriate business strategies. In recent years, we have witnessed explosive growth in the usage of social media. With four hundred million social media users within India, these channels, especially Twitter, have become a ubiquitous and dominant platform for social networking, content sharing, and marketing. Customer-to-Firm and Customer-to-Customer interactions through social media are creating significant opportunities for all kinds of organizations. But it comes with its own set of challenges. Social interactions are difficult to understand due to the extensive usage of slang, code-mixing, virality, over-representation of the top-tier elite population, different kinds of promotional content from brands, third-party sellers, and marketplaces. Many organizations run initiatives as well to increase their social reach. Only two or three mentions out of every ten are genuine feedback from the customers expressing their experience with a particular platform or product. And if gathered accurately, these could be a super valuable source to understand the overall sentiment of the same.

To measure the customer experience, we use a well-established metric across organizations called Net Promoter Score (NPS) [12, 19]. Mathematically, it is defined as %promoters - %detractors. In short, one can measure the customer loyalty and satisfaction by asking customers how likely they are to recommend any product or service to others on a scale of 0-10. As a business metric, it can help companies of all sizes organize around an important goal - increase their score by earning more enthusiastic customers - that can be easily tracked and quantified over time. It is a valuable strategic metric, but the score is not enough by itself to paint a complete picture. Below are a few of the challenges associated with it. **1)** Running the survey could be expensive, **2)** For a given score, we might not have insights into the actual pain point of the customers. **3)** Conventional surveys handle the last problem by asking follow-up questions, but many customers don't prefer to participate unless incentivized. Hence, many questions could still go unanswered.

To handle the above bottlenecks, we present our research on building a generic framework to retrieve valuable information from the ocean of Social Media and estimate NPS using it. The same can be leveraged across different institutions if calibrated carefully. We divide the complete pipeline in two stages: **1)** We pass the raw crawled data through our Data Cleaning pipeline, which filters out the viral content and initiatives and applies the appropriate tier-adjustment scaling. **2)** We then use a T5 model fine-tuned on carefully annotated 15K social media mentions collected from diverse channels like Twitter, Facebook, Instagram, and YouTube (by e-commerce experts) to identify the ten different customer

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524219

journey nodes, if present, with their respective sentiments. Lastly, we do feature engineering and pass it through our feature selection pipeline to train an XGBoost model.

2 DATA CLEANING PIPELINE

2.1 Data Source

We use a third-party tool to extract the mentions from the previously mentioned social media channels. It asks the users to input a list of keywords (different spellings of a marketplace in our case) and uses a simple keyword-based case-insensitive regex in the backend to find all the posts where it gets a hit. For instance, keywords for Flipkart could be - Flipkart, Ekart, Flip kart. Each post is associated with six different fields. **1) UnivesalMessageId:** Unique identifier of each post, **2) Message:** Content of the message, **3) SocialNetwork:** Name of the channel, **4) MessageType:** Tells the exact source (Direct Message, Reply, Comments, etc.,) of a particular channel, **5) Company:** Name of the MarketPlace, and **6) Location:** Location of the user, if available in the public profile.

2.2 Different Challenges

1) MessageTypes filtering: There are some message types where volumes differ significantly across marketplaces or are exclusive to a particular marketplace. If we are performing any comparison with our competitors, we remove these to ensure fairness. We also found a few message types that are not the voice of the customers (VoC) per se and only contain Customer Care agent responses to the queries raised.

2) Viral Content Removal: More often than not, posts go viral on social media, and within a small timeframe, they get reposted multiple times. To reduce the over-indexing on a few messages, we use a heuristic approach to apply a max cap on them. Doing this will make sure that a single type of message doesn't impact the VOC metrics significantly. For instance, this tweet has 1000+ retweets - *USERNAME No action against MARKETPLACE for daylight violation of policy rules, but traders are quickly penalized even for a small mistake. For how long will traders suffer? USERNAME #purifyecommerce URL*. These kinds of messages can come from any user and not just popular figures on the internet. Many times users just append emojis or make a small change to the text (Twitter Retweet, for example), so, doing an exact regex match doesn't solve the problem completely. Hence, we rely on this elegant approach released by Google [5] a few months back for better de-duplication.

3) Tier Adjustment: We observe that users from non-metro cities are less likely to post on social media about their experiences (in context of e-commerce) than from metro cities. Hence, to get a fair representation, it is crucial to give more weight to messages coming from non-metro cities. We use a simple weighing strategy, depending on the source location of the mention to handle it.

$$\mathbf{Wt} = \frac{\{Metro/Non\_Metro\}\_Gross\_Merchandise\_Value}{\{Metro/Non\_Metro\}\_Total\_Mentions} \quad (1)$$

4) Initiative Identification: Many brands or marketplaces try to increase their reach by giving some incentive to their users to tweet positive things about them. It is crucial to remove these messages as these are not genuine VoCs, and users write them only in the hope of getting rewarded. Let's take an example of this - *Looking out for amazing surprises this XXXX Day? Then screenshot the correct XXXX Day dates, retweet it along with the #DiscoverJoy, and stand a chance to win an amazing vouchers!.* We can see that the marketplace is giving vouchers as rewards to users for sharing the screenshot with #DiscoveryJoy hashtag in the tweet.

The main challenge in identifying such initiatives and responses to these initiatives is that these posts generally lack context (sometimes just a screenshot). Replies from customers to these posts can also be identical to genuine feedback. Sometimes, valuable information is encoded just in the hashtags. But a few of them can be generic, and the users might start using them in an entirely different context after some point in time. We observed in our study that these initiatives and responses to them hold a few properties like **1)** While introducing any new initiatives, marketplaces request the users to use atleast one particular hashtag in the text via the Twitter channel. **2)** Responses to these initiatives almost always contain that hashtag and generally have sentiment skewed in the positive direction as compared to other VoCs. **3)** Most of them are valid for less than a month or so. So, we have built an iterative algorithm using these insights to remove these messages with almost **85%** F1 score (very high precision, but relatively small recall).

3 MACHINE LEARNING PIPELINE

Once we have the clean crawled data, we pass it through our Machine Learning pipeline, which has **T5** [11], a transformer-based [15] model, at its core. We went ahead with T5 primarily for three reasons. **1)** An efficient way to do multitask, multistage training, where we can share the same decoder head across almost all Natural Language Processing (NLP) tasks, **2)** T5 is one of the best performing models, with results close to recent state-of-art on public leaderboards [16, 17], and **3)** Tasks that we are dealing with fit very well in the text-to-text framework, and significantly simplifies our data loading pipeline.

To extract actionable insights, we curated a list of aspects that captures the complete lifecycle of the e-commerce experience. We can broadly categorize them into two buckets - **1) pre-delivery** nodes like Selection, Pricing, Payment, and Cancellation, and **2) post-delivery** nodes like Customer Care (also a pre-delivery node), Delivery, Product Quality, Return, Installation, and Loyalty Program. We have an Aspect-based Sentiment Analysis (ABSA) model to extract all the relevant aspects with their respective sentiment. We observe that this approach works very well in practice to measure the perception of the overall e-commerce platform.

3.1 Aspect-based Sentiment Analysis

Aspect-Based Sentiment Analysis and Targeted ABSA (TABSA) aims to identify fine-grained consumers' opinions about different aspects of products or services. Analyzing the language used in a review is a difficult task that becomes even more difficult for TABSA and requires a deep understanding of the language. With the advent of self-attention mechanism, deep language models, such as BERT [4], RoBERTa [7], T5 [11], GPT-3 [1], have shown substantial progress in this regard. Many recent ABSA papers have shown significant performance gains by using BERT-like models

either as an embedding layer [6, 14, 20] or fine-tuning them with a classification head [13, 21].

We frame our current problem as a multi-label problem, and the following is the reasoning behind it. We have a predefined list of ten aspects, and all can take any of the three sentiments - Positive, Neutral, and Negative. So in total, we have thirty possible labels of the format {aspect-sentiment} (e.g., delivery-positive, pricing-negative), and theoretically speaking, any of them can co-occur with any other. Since this task is specific to our use case, we start with the dataset tagging process described in the next section.

3.2 Dataset Preparation

With 3.6B social media users worldwide, social media is also a predominant channel for marketing by brands (like Samsung, Apple, Oppo), third-party sellers, and marketplaces (e.g., Facebook, Instagram, Flipkart, Amazon). We found in our experiments that roughly 70% of the data doesn't provide any meaningful insights and are just different kinds of promotion or initiatives. So, to have a well-balanced dataset that we could start labeling and not waste too much time on promotional content, we came up with a heuristic list of keywords for every aspect that acted as weak supervision in a way. This strategy helped us create a dataset of size 20k with approximately 10% intentional volume from the "Untagged" category, and the rest 90% among all others. The keyword-based model is by no means very accurate but provides us a good starting point in alleviating a skewed distribution of labels in the training dataset.

We tag the dataset in two iterations with two different objectives in mind - experimentation and scaling. The goal of the first iteration was to start with a small subset of data (around 3k examples) and refine our initial set of rules with an appropriate owner-tagger feedback mechanism in place. It was a hectic part but played a very crucial role in creating a gold-standard dataset. In the second iteration, we just scaled it up and tagged the remaining data points.

3.3 Labeling Accuracy

We asked the annotators to tag the top five ABSA tags, which could mean dropping a few aspects with the neutral sentiment. We always share a batch of data with two annotators first. Since this task is subjective, we allow a tolerance of 1, so the list of tags from both the annotators can have a difference of 1 in length, but the rest should exactly match each other. At this stage, we discard all the data points where the mismatch is more than one and share the data again with the third annotator. We follow the same process as above (i.e., if labels generated by the third annotator match with any last two sets within a tolerance of 1, we include them in the final dataset, otherwise, we remove them). Here is a quick summary.

- On **20k** examples that we started with, the Exact Match Score (EM) with a tolerance of 0 (i.e., labels from the first two annotators exactly match each other) is **55.55%**, and a tolerance of 1 is **67%**. At this step, we discard all the data points where the mismatch is more than one.
- On 6.5k examples, where mismatch is more than 1 in the last step, the EM with a tolerance of 0 (i.e., labels from the third annotator exactly match with any of the previous two) is 55.55%, and a tolerance of 1 is 67%. We finally discard the remaining 3k out of 18k examples, where none of the sets match each other within a tolerance of 1.

3.4 Training Experiments

We use the T5 model and fix the maximum source length to 256 in all the experiments. For sentences containing more than 256 tokens, we apply a rolling window approach with 20% overlapping context and finally merge the predictions. Here is a small summary of a list of experiments that we performed.

- **Vanilla Fine-tuning:** We finetune the small and base variants of T5 with a constant learning rate of 5e-4, batch size of 64 (maximum that we could fit), and gradient accumulation step size of 2. We came up with this set of hyper-parameter after running a grid search over a small range for the learning rate and gradient accumulation step size.
- **Oversampling:** We observed that even though we tried to maintain uniformity at the aspect level before labeling, the final distribution was somewhat skewed. We also saw an imbalance in sentiment, and we attribute this to the fact that most of the time, customers post on social media only when they are dissatisfied with the services. Hence, we apply a heuristic multi-label oversampling strategy to make the distribution less skewed. It has been shown in the literature that simple oversampling [2] works very well with CNN for imbalanced multi-class classification problems.
- **Multi-stage Fine-tuning:** Pre-training on a few related auxiliary tasks could substantially improve the performance of the final downstream task [8, 9]. Since our dataset was already small to train a good ABSA model, we introduced two auxiliary tasks - Aspect and Overall Sentiment identification to see the impact on the final ABSA performance. To get the data at the aspect level, we drop the sentiment-related information from the labels. For the sentiment, we just go ahead with overall sentiment. Now the training of these additional tasks can happen in one of these ways: **1)** Aspect –> Sentiment –> ABSA, **2)** {Aspect, Sentiment} –> ABSA, and **3)** {Aspect, Sentiment, ABSA}, where tasks grouped under {} mean that they are trained together. Since T5 supports multi-task learning, we explore all the options listed above. For the last two configurations, we train to maximize the average F1 score. For the last configuration, we also check the performance after finetuning just on the ABSA task again.

3.5 Results

Table attached below shows an overview of the performance of the different experimental setups. The metrics that we use for the evaluation are: **1)** F1-score, and **5)** Exact Match Score. EM assigns a hard penalty to any kind of mismatch, but the F1 score gives a partial score, depending on the recall and precision. We find that training on auxiliary tasks and fine-tuning on ABSA again does help and gives the best performance. To our surprise, oversampling hurts performance and leads to overfitting, even though we ensured that all labels appear with almost same frequency in a single batch.

	t5-small		t5-base	
	F1	EM	F1	EM
ABSA	79.48	62.71	80.72	64.69
ABSA + Oversampling	79.11	62.68	80.45	64.1
{ACLF} - {SENT} - {ABSA}	79.50	63.56	80.96	66.25
{ACLF, SENT} - {ABSA}	79.27	63.65	80.9	65.94
{ACLF, SENT, ABSA}	79.67	63.67	81.11	66.5
{ACLF, SENT, ABSA} - {ABSA}	**79.89**	**64.17**	**81.69**	**67.19**

Table 1: F1 and Exact Match score of t5-base and t5-small models in different configurations.

4 NET PROMOTER SCORE

4.1 Data Source

Just after an order is delivered, customer receives digital survey through emails or in-app questionnaires and their responses are captured near real-time. We aggregate these responses to find weekly NPS and use it as the target variable in our training pipeline. To generate the input features (as described in detail in the next section), we run inference on the VoC data using the ABSA model and aggregate the mentions at week level for each {aspect-sentiment} tag. In total, we have data points for 130 weeks, which we split into 100, 15, and 15 (almost a quarter of a year) for training, validation, and test set.

4.2 Feature Engineering

We synthesize broadly three kinds of features for every aspect at the organizational level. **1) Sentiment vars**: An indicator of the overall sentiment of that aspect and is calculated similarly to NPS, i.e., %Positive_Aspect_Mentions - %Negative_Aspect_Mentions. **2) Mention Ratio vars**: Since a few nodes are biased towards a particular sentiment, it doesn't give us a complete picture. So, we use the Mention share of that aspect wrt total mentions to capture those relations. **3) Weighted Sentiment vars**: It is simply the product of Sentiment and Mention Ratio of that aspect. We found in our empirical analysis that combining both features, i.e., Weighted_Aspect_Sentiment features, retains all the relevant information and boosts the final performance as well. For example, for a particular week **w** and aspect **asp**, the above features are calculated as follows:

$$Asp\_Sent[w] = \frac{Pos\_Asp\_Ment[w] - Neg\_Asp\_Ment[w]}{Total\_Asp\_Ment[w]} \quad (2)$$

$$Asp\_Ment\_Ratio[w] = \frac{Total\_Asp\_Ment[w]}{Total\_Platform\_Ment[w]} \quad (3)$$

$$Wtd\_Asp\_Sent[w] = Asp\_Sent[w] \times Asp\_Ment\_Ratio[w], \quad (4)$$

where Ment, Sent, Pos, and Neg stand for Mentions, Sentiment, Positive, and Negative, respectively. On the feature transformation side, we create a first-order lag variable for every feature and take the log-transformation of the positive ones, and finally append them to our initial feature set. We represent the VoC of a week as a vector of all the generated features, calling it x[w]. The target variable for this week is the NPS, calling it y[w].

4.3 Feature Selection

- **Statistical Methods:** We start by dropping irrelevant features based on our business understanding. We keep only one of the two features - original (feat) or ln (feat), depending on which has better mutual information with the target variable. In case there is no ln transformed variant, we skip this step. We finally drop highly correlated features with the threshold set to 95%.
- **Machine Learning Wrappers:** We run forward selection based on sorted permutation importance and choose the one with the best validation results.

4.4 Results

We use XGBoost [3] model for all our experiments (even for the feature selection), which is a decision-tree-based [10] ensemble ML algorithm that uses a gradient boosting [18] framework. We perform a random search over three hyperparameters (max_depth, n_estimators, and subsample fraction) for the optimal performance. We use two metrics to evaluate our model, namely Pearson Correlation Coefficient (PCC) and Mean Absolute Percentage Error (MAPE). The current model has 1.9% MAPE and 79% Pearson correlation on the test set. **Figure 1** shows the performance of the model (y-axis and x-axis labels have been masked, and data is also shuffled to modify the chronological order for privacy reasons).

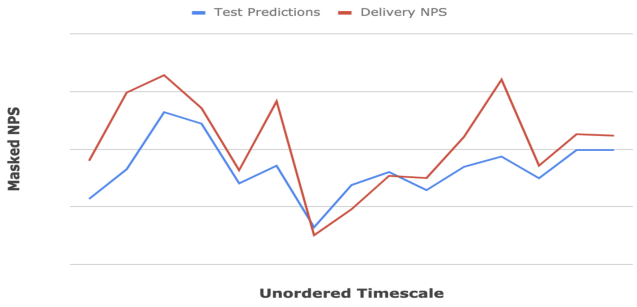

Figure 1: Masked Delivery NPS vs Predictions (not in chronological order)

5 CONCLUSION

The quality of the conventional methods depends heavily on the questionnaire asked in the survey and customers' response bias, as well as we believe that people in the current era express themselves more freely on social media when they have bad experiences with a product or platform. This paper presents the research on estimating the NPS via mentions extracted from the major social media channels, showing that it is more efficient and easily scalable than traditional approaches. The ability to extract the aspects with their respective sentiment, which gets enabled by our ABSA model, accurately plays a significant role in forecasting NPS and finding the actual drivers. But, this approach has a few limitations as well, which we leave for future research: **1)** social biasness (which includes low penetration in tier 2+ cities, virality, platform initiatives, and negative sentiment), **2)** a better oversampling strategy for multi-label classification, and **3)** an efficient way to do targeted-ABSA with small language models.

REFERENCES

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL]

[2] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks* 106 (Oct 2018), 249–259. https://doi.org/10.1016/j.neunet.2018.07.011

[3] Tianqi Chen and Carlos Guestrin. 2016. XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Aug 2016). https://doi.org/10.1145/2939672.2939785

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

[5] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2021. Deduplicating Training Data Makes Language Models Better. *arXiv preprint arXiv:2107.06499* (2021).

[6] Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019. Exploiting BERT for end-to-end aspect-based sentiment analysis. *arXiv preprint arXiv:1910.00883* (2019).

[7] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[8] Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. 2021. What to Pre-Train on? Efficient Intermediate Task Selection. arXiv:2104.08247 [cs.CL]

[9] Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-Task Transfer Learning with Pretrained Models for Natural Language Understanding: When and Why Does It Work? arXiv:2005.00628 [cs.CL]

[10] J. Ross Quinlan. 1986. Induction of decision trees. *Machine learning* 1, 1 (1986), 81–106.

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

[12] Frederick F Reichheld. 2003. The one number you need to grow. *Harvard business review* 81, 12 (2003), 46–55.

[13] Emanuel H Silva and Ricardo M Marcacini. [n. d.]. Aspect-based Sentiment Analysis using BERT with Disentangled Attention. ([n. d.]).

[14] Youwei Song, Jiahai Wang, Zhiwei Liang, Zhiyue Liu, and Tao Jiang. 2020. Utilizing BERT intermediate layers for aspect based sentiment analysis and natural language inference. *arXiv preprint arXiv:2002.04815* (2020).

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

[16] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537* (2019).

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

[18] Wikipedia contributors. 2021. Gradient boosting — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Gradient_boosting&oldid=1055694616 [Online; accessed 18-November-2021].

[19] Wikipedia contributors. 2021. Net promoter score — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Net_promoter_score&oldid=1054323643 [Online; accessed 17-November-2021].

[20] Zhengxuan Wu and Desmond C Ong. 2020. Context-guided bert for targeted aspect-based sentiment analysis. *Association for the Advancement of Artificial Intelligence* (2020), 1–9.

[21] Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv preprint arXiv:1904.02232* (2019).