Pattern Product Review Analysis

Technical Report - Milestone 1

Team: Sehaj Chawla, Taro Spirit, Lotus Xia, Heather Liu

Outline

1. Problem Description

eCommerce is gaining dominance in the global retail market. Due to the lack of in-person interactions on virtual marketplaces, customer reviews become one of the most important channels for customers to communicate with each other and to provide feedback to retailers.

On the one hand, customers rely on reviews from past customers to have a peek at product quality and to make purchasing decisions accordingly. On the other hand, understanding customer reviews is also crucial to the success of online retailers. For instance, retailers can improve product quality or change marketing strategies based on customer preferences and suggestions. Retailers may also rely on review sentiment to predict sales volumes in the immediate future and to determine the quantity of inventories to acquire.

In this project, we worked with Pattern, an eCommerce accelerator, to create a natural language processing (NLP) model that 1) predicts future sales volumes from customer reviews and 2) extracts salient, predictive features from these reviews. The predictive power of reviews will be compared against a regression model solely using the variables other than the review text data. Also, the performance of NLP models will be compared with a bag-of-words model.

2. Data

The BSR history dataset is a history of Best Seller Rank (BSR) for Amazon products within the

Vitamins and Dietary Supplements category, ranging from July 2017 to July 2021. BSR is a performance evaluation calculated by Amazon using the product's current sales volume as well as the product's historical sales. Rank 1 in a category is the best selling product, 2 is the second best, etc. This dataset contains around 29 million rows and covers 9,991 unique products. The dataset's fields include ASIN (a unique identifier for the product), best seller rank, the average price of the product over the past 180 days, and the date of observation.

The Review history dataset is a collection of Amazon reviews for products within the Amazon Vitamins and Dietary Supplements category spanning from July 2017 to July 2021. There are around 5 million reviews on 9,977 unique products. The dataset's fields include ASIN (a unique identifier for the product), product name, review title, review rating, review date, review votes, review comment count, and a binary field indicating whether the purchase is verified. The two datasets cover 9,958 products in common.

The original datasets have two main limitations. Firstly, the time span is not equal across products. More popular products usually have more observations than less popular products. Therefore, the samples in our dataset may not be representative of the true population of products and reviews on Amazon. In addition, the dates at which reviews were collected do not necessarily match the dates at which we know the rank of the respective product, sometimes leading to a small overlapping period for us to work with. Second, the ordinal nature of the BSR hides some important and relevant information we may care about as an online retailer. For instance, the sales volume of rank 1 may differ drastically from the sales volume of rank 2, although they only differ by 1 in terms of ranking. In addition, Amazon does not reveal how exactly they calculate the BSRs. It appears to be calculated based on mostly the daily sales, with little weights on historical sales of the product. As a result, the ranks may suddenly spike or drop, adding noise to the BSR data.

3. Exploratory Data Analysis

Figure 1 plots the BSR over time for an example product (ASIN: B00005313T). The raw BSR data is sometimes volatile. We note two major causes of the volatility: First, some products may go out of stock for a short period of time, during which the BSR of said product skyrockets. Second, products may go on sale, leading to a higher sales volume, and, in turn, a lower BSR during the sales period. The main focus of this project is not to predict day to day spikes and drops that happen due to these external factors. It also appears unlikely for us to predict granular rank movement using just review data. Therefore, we choose to aggregate product ranks by taking the median rank per month for each product. We discuss more about the pre-processing procedures below.

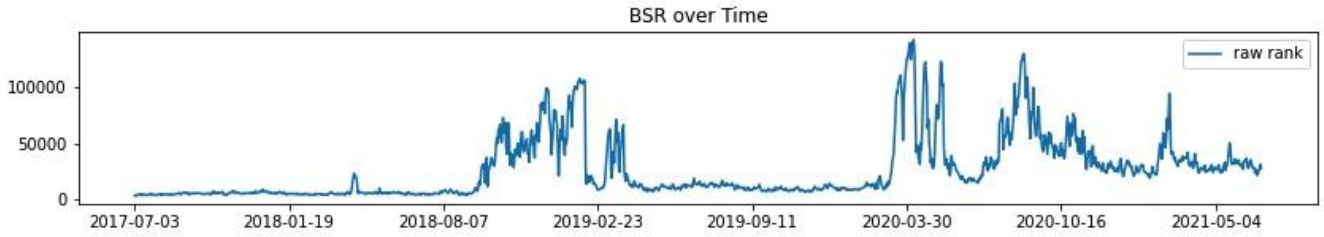

Figure 1: BSR over time for product B00005313T

Similarly, we aggregate the review data by product and month. The number of reviews per product-month varies quite drastically across products. The left pane of Figure 2 presents a histogram of the log number of reviews per month. The population is right skewed, with the most popular product attracting 1392 reviews per month. Each review also has different lengths. The right pane of Figure 2 plots the histogram of the log number of words in each review. While approximately 28% of reviews have fewer than 10 words, the longest review contains 4643 words.

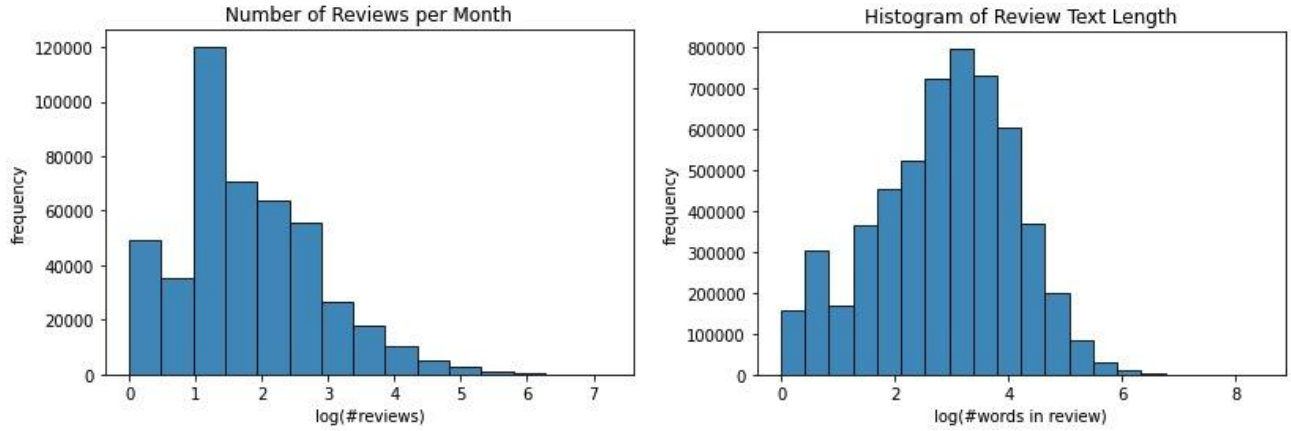

Figure 2: Histograms of the number of reviews and length of reviews

Among all review metadata, review rating is one of the most salient features that influence customer decision. Figure 3 presents some high-level statistics by review ratings. The vast majority of reviews are 5-star reviews. A potential explanation for this pattern is that popular products are more likely to elicit reviews, and these products are popular because they have good quality. In addition, reviews with 5-star ratings tend to be shorter than the reviews with fewer stars. 2-star reviews are the longest on average, which is intuitive—reviewers may want some lengthy explanations to justify their low ratings.

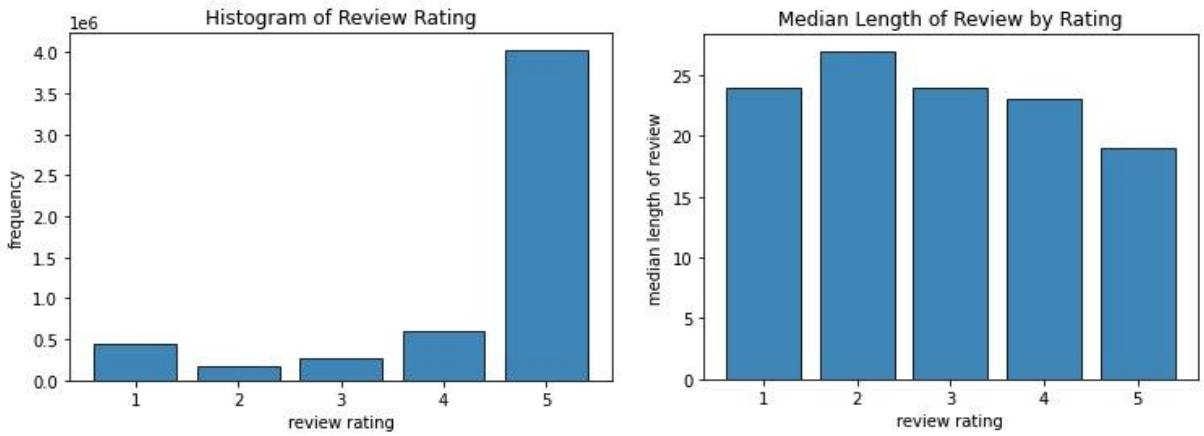

Figure 3: Frequency and review length by review rating

Lastly, we examine the number of upvotes corresponding to each review. Reviews with different numbers of votes are expected to have different impacts on customer decisions. In particular, the most voted review is displayed on top of the review feeds, so more customers will notice and trust this review. Figure 4 plots the distribution of the log number of upvotes per review. The distribution is again right-skewed. While over 60% of reviews have zero upvotes, the most liked reviews are voted by 16,368 customers. Despite its indisputable impact on purchasing decisions, we would like to flag that the number of upvotes may be a form of data leakage. These votes are a snapshot as of the time of scraping—using these numbers to predict sales performance in the past may seem inappropriate.

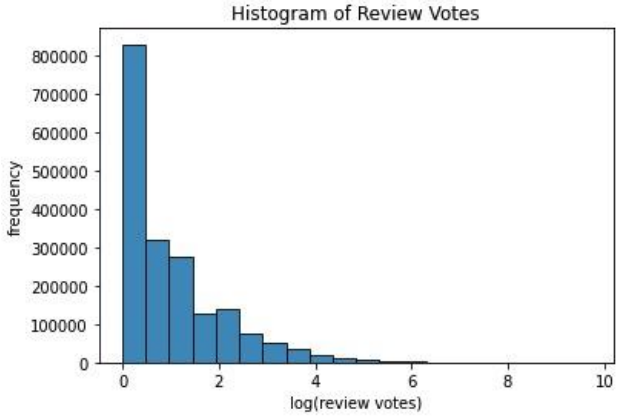

Figure 4: Histogram of the number of review votes

4. Data Processing

We take the following steps to pre-process the BSR:

1. We first grouped the data by product, yielding the historical ranks with corresponding dates stored in lists.
2. We then excluded products with more than 30 days of consecutive missing BSRs. For the

remaining products, we filled in the missing values by taking the median of the BSR right before and after the missing range.
3. To reduce the noise in the data, we smooth the BSR data by calculating a 10-day rolling median.
4. For the training purpose, we then grouped the data by month and calculated the median BSR for each month.
5. Lastly, we normalized BSR for each product against its best performance, i.e. for each product, we first computed its maximum and minimum BSR observed in the dataset; then, we used the Max-Min method to scale down the value of BSR to between 0 and 1. We reshape the data to yield a product-month level dataset.

For ease of reference, in the rest of this report, we refer to the monthly median BSR as "monthly BSR" and the rolling median as "rolling BSR."

We take the following steps to pre-process the review data:

1. We generated reviewvotes_num which is a numeric version of the number of votes for each review.
2. Then we grouped reviews by product-month. Texts/values in each product-month level are grouped in a list. We then merged the dataset with the processed BSR dataset to keep the products in the intersection.

5. Literature Review

Recent work has demonstrated substantial gains from applying NLP models on texts for solving broader business problems, such as prediction of sales volume, sentiment analysis, and market segmentation. In [1], the authors used large-scale text mining to characterize the behavior of e-commerce consumers and modeled the relationship between product presentations and business outcomes. Also, as described in [2], the authors established a feature-level sentiment analysis using Amazon sales data and customer review data, to show that customers' preference for different features and texts can be used for predictive modeling of future changes in sales. In [3], the authors presented a sentiment-aware model for predicting sales performance using blogs.

In most of the papers, Transformers, proposed in [4], is widely used for text encoding. Transformers is an encoder-decoder-based neural network whose main feature is the use of so-called p (i.e., a mechanism that determines the importance of words to other words in a sentence or which words are more likely to appear together) and the absence of recursive connections (or recurrent neural networks) to solve tasks involving sequences (or sentences). Especially, BERT, one of the transformer family, is a state-of-the-art machine learning model used for NLP tasks. It is a way of learning representations of a language that uses a transformer, specifically, the encoder part of the transformer. It is widely used for text representation as well.

6. Models

We have started by working on two regression models

(1) a model without any review text in the input and only uses review metadata, and

(2) a model based almost completely on review texts.

The idea behind this approach is to look at the predictive power of the text as an incremental

value-add on the model without any text. We plan to do this by ensembling our two models, once we are satisfied by their performance individually. This would help us with the interpretability of the models (our second task), while also giving valuable modeling insights into our prediction task. Below, we describe our current progress with the two models, as well as future work we have planned.

6.1 Regression without Text

We consider linear regression, xgboost, and random forest models to predict the future change in rank of a product based on most of the data we have available apart from the text of the reviews and their titles. For every product, we predict the change in monthly BSR using the following features:

1) The rolling BSR.
2) The monthly BSR.
3) The monthly mean of the product's review ratings.
4) The monthly mean of the product's review ratings, weighted by the number of upvotes for each review.
5) The monthly mean of the product's review ratings, weighted by the truth function associated with the verification of the reviews, i.e. weight a review by one if it is verified and weight by zero if it is not verified.
6) The average over all the monthly product's review ratings mean of all the previous months, i.e. the average over the values of point 3 calculated for every month.
7) The average over all the means of the product's review ratings over the previous months, weighted by the number of upvotes, i.e. the average over the values of point 4 calculated for every month.
8) The average over all the means of the product's review ratings over the previous months, weighted by the verification truth functions, i.e. the average over the values of point 5 calculated for every month.

In total that represents 37 features. The last three features are cumulative datas which give the model the information about the product's past performance, instead of having only the information about the current month. This is important as customers have access to all of the reviews which were ever posted for that particular product. In particular, the overall rating visible to the customer is computed based on all of the reviews for that product. Moreover, the first reviews which are visible to a customer are the reviews which have the most upvotes and we therefore weight those reviews more.

We then input these 37 features into linear regression, xgboost, and random forest models imported from sklearn. Training on 65 products (approximately 2000 months overall) and predicting for a test data set of 5 products (approximately 150 months), the overall performances of the different models are:

- Linear regression:
    - Mean squared error: 0.0155
    - R-squared score: 0.2222
- Xgboost:
    - Mean squared error: 0.0138
    - R-squared score: 0.311
- Random forest:
    - Mean squared error: 0.0134

- R-squared score: 0.3307

It is impressive that 33% of the variance is already explained by this random forest model as we are training on a very small dataset and most of the variance in the data is expected to be hard to explain. Indeed, as seen previously in Figure 1 the BSR data varies a lot and is therefore expected to be hard to explain even after preprocessing of the data.

6.2 Regression with Text

Our model with text, uses transformer embeddings of reviews as inputs, along with the monthly BSR for the previous month, and the number of votes each of those reviews got. It then tries to predict a regression based target of the change in median BSR from the current month to the next. Below, we explain our current model structure with the help of the diagram in Figure 5.

The three inputs we have for the model are:

1. a list of review texts (labeled `review_text`),
2. a list of corresponding review upvotes each of the reviews received (labeled `votes_input`), and
3. a list of 30 numbers that represent the moving median BSR for the 30 days in the month prior to the prediction (labeled `bsr_input`).

First, we create the transformer embeddings for each of the reviews (in this case using large uncased BERT), and change the vector dimensions to 400 (an arbitrary value that we will experiment with later). This makes sure that no matter which transformer we use, the latent dimensions are consistent. We then weigh each of the embedding vectors by the number of votes that review received. Our driving hypothesis here is that the reviews with higher upvotes are the ones that are at the top of the amazon buyer's page, and are thus the most powerful in predicting performance for the future. So, we end up with a final latent description of the reviews, which represents a weighted average across the embeddings created by the transformer (the weights being the number of votes the review received). This is represented at the layer labeled as `agg_review_repr`. This structure also makes the model robust against the changing number of reviews in each row, since the number of reviews a product receives in a month are very different depending on the product and the month.

We then concatenate this processed information with the total number of upvotes across all reviews for this product, and the moving median BSR (the concatenated layer is labeled as `concat_all_inputs`). Finally, we pass this layer through another couple of dense layers to get our regression value. Moreover, we have used multiple dropout layers to help with reducing model variance.

This model is relatively complicated with a large number of trainable weights, and therefore, in the time and resources we have had so far, we have only trained it on 200 rows out of the total 250,000 rows we have access to. The performance of the model after this very limited training has the following metrics:

- Mean squared error: 0.0504

7. Future Work

We plan on experimenting with a few structural choices we have made based on our internal discussions and Pattern's suggestions during our meetings:

For the regression model without text:

1. Include additional features such as the number of reviews, the ratings of the five most upvoted reviews, price data, etc.
2. Hyperparameters optimization.
3. Train and test over the whole data set (approximately 250k months) on AWS.
4. Feature importance analysis, e.g. how predictive is the number of reviews for a product.

For the regression model with text:

5. Look at just the current month's reviews as inputs rather than looking at all the reviews so far cumulatively (which is what we do currently).
6. Changing our transformer from BERT large to tiny BERT to improve our utilization of resources while still maintaining the quality of results.
7. Pattern also mentioned that the upvote count that we have access to are the upvote values at the time of scraping, and so they are not the upvote counts at the month we are trying to predict during. This is, therefore, a form of data leakage, and we are thinking of ways of reducing the impact of this by potentially not weighting by upvote count, but just taking an unweighted mean across all reviews.
8. We also plan to increase our train and test size once we have access to more cloud resources like AWS or Azure.
9. Training a bag of words model to confirm whether BERT outperforms simpler linear models, while also improving our ability to interpret model performance.

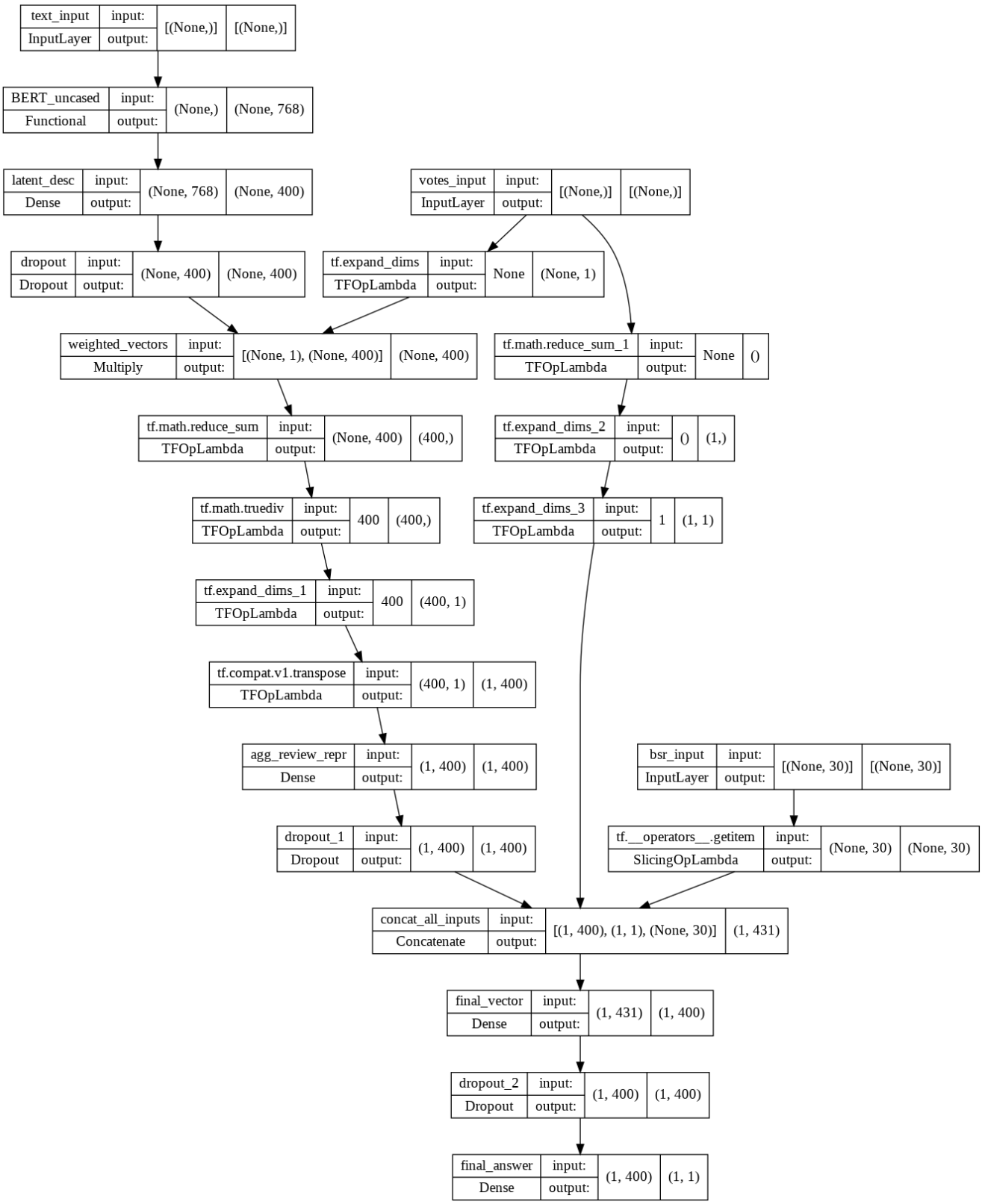

Figure 5: Structure for Model with Text

References:

[1] Pryzant, Reid, Young-joo Chung and Dan Jurafsky. “Predicting Sales from the Language of Product Descriptions.” eCOM@SIGIR (2017).

[2] Nikolay Archak, Anindya Ghose, Panagiotis G. Ipeirotis, (2011) Deriving the Pricing Power of Product Features by Mining Consumer Reviews. Management Science 57(8):1485-1509.

[3] Yang Liu, Xiangji Huang, Aijun An, and Xiaohui Yu. 2007. ARSA: a sentiment-aware model for predicting sales performance using blogs. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '07). Association for Computing Machinery, New York, NY, USA, 607–614.
DOI:https://doi.org/10.1145/1277741.1277845

[4] 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 Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.