IACS Harvard School of Engineering and Applied Sciences
IACS Institute for Applied Computational Science

Pattern Product Review Analysis

Technical Report - Milestone 2

Team: Sehaj Chawla, Taro Spirig, 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 work with Pattern, an eCommerce accelerator, to create a natural language processing (NLP) model that 1) predicts future sales ranks/volumes from customer reviews and 2) extracts salient, predictive features from these reviews.

For task 1, we compare the performance of models using only review metadata (everything but review texts) and the performance of models using only review texts. We would like to evaluate the size of improvement, if any, from using review texts in addition to review metadata. For task 2, we hope to generate a dictionary of keywords or phrases that are predictive of sales performance that makes intuitive sense and are informative for online retailers in terms of future sales strategy. These tasks remain at an exploratory stage at the moment—we will gauge the model performance and formulate a success metric as we move forward.

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. The data is scrapped by a third-party service provider, Keepa. 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 that are on sale at the time of scraping.[1] The earliest review is in Jan 2004 and the last review in this dataset is in [TODO]. There are around 5 million reviews on 9,977 unique products. The dataset's fields include ASIN, product name, review title, review rating, review date, review upvotes, review comment count, and a binary field indicating whether the purchase is verified (An "Amazon Verified Purchase" review means Amazon has verified that the person writing the review purchased the product at Amazon and didn't receive the product at a deep discount). The two datasets cover 9,958 products in common.

We note two main limitations of the datasets. Firstly, the time span is not equal across products. Keepa, the third party data provider, pays more attention to popular products on Amazon, scraping and updating their ranking information more often than unpopular products. Therefore, popular products are more likely to have frequent, continuous BSR information in our dataset than unpopular products, which tend to have long periods of missing BSR. As a result, when we restrict to products that have frequent reviews and BSR records, the final sample may not be representative of the true population of products and reviews on Amazon. In addition, the dates at which we know the product ranking may not match the dates at which reviews were added for said product, sometimes leading to a small overlapping period for us to work with.

Secondly, 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. The calculation seems to depend most heavily on 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.

To address the second limitation, Pattern further provides us with their estimated sales volumes corresponding to each BSR. The mapping is fitted based on actual sales volumes and ranks of their customer retailers, although the exact mapping function between BSR and sales volume is not revealed to us. A drawback is that the estimate for top products may not be accurate, since the Pattern's customers rarely attain an extremely small rank (e.g. below 100). The estimates on these top products are essentially based purely on extrapolation of the fitted function.

3. Exploratory Data Analysis

3.1. BSR

[1] Around fall 2021.

Figure 1 plots the BSR over time for an example product (ASIN: B00005313T). The raw BSR data is oftentimes volatile. In particular, the median difference between the highest and lowest BSR we observe for a product is over 170,000. The least volatile product has a difference of 474 between the highest and lowest ranking, whereas the most volatile product has a difference of 7785554.

We note three major causes of the volatility:

First, the product scope based on which the ranks are computed is larger than the scope of products we have in our datasets. In particular, our datasets consist of all products in Amazon's Vitamins and Supplements category, but the ranking is calculated based on all products in Amazon's Health and Household category, which is a much larger universe. As an example, the largest rank we see on a day may be on the scale of millions, although we only have data on approximately 9000 products in our dataset.

Second, some products may go out of stock for a short period of time, during which the BSR of said product skyrockets.

Third, products may go on sale, leading to a higher sales volume, and, in turn, a lower BSR during the sale 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 chose to aggregate product ranks by taking the median rank per month for each product. We discuss more about our pre-processing procedures below.

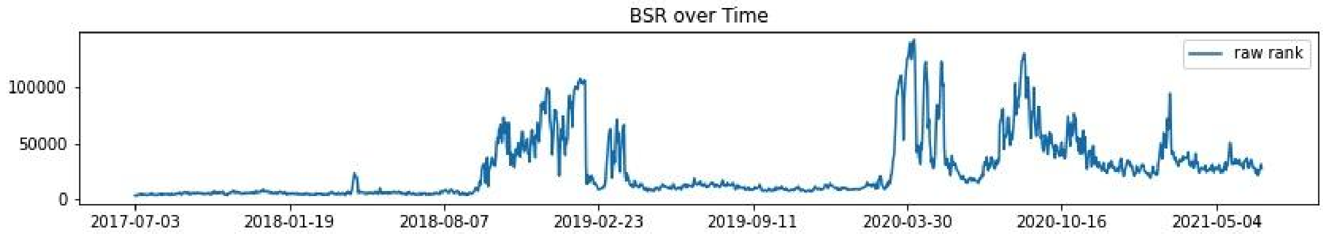

Figure 1: BSR over time for product B00005313T

3.2. Sales Volumes

Figure 2 plots the estimated sales volume corresponding to each BSR. We note that the sales volume decreases quite rapidly as BSR increases. Rank 1 product is estimated to have a daily sales volume of 5,214 units, whereas the sales volume of rank 7000 is barely above 100 (not shown in the plot), and the sales volume of rank 201934 dips below 1. The smallest sales volume in the dataset is at 0.05 units for a BSR of 454,302. As a comparison, the largest BSR observed in our dataset is on the order of 3 million, which is much larger than the largest BSR associated with an estimated sales volume. However, these products with extremely small ranks have essentially zero sales—we later describe a simple extrapolation method in "data processing" section.

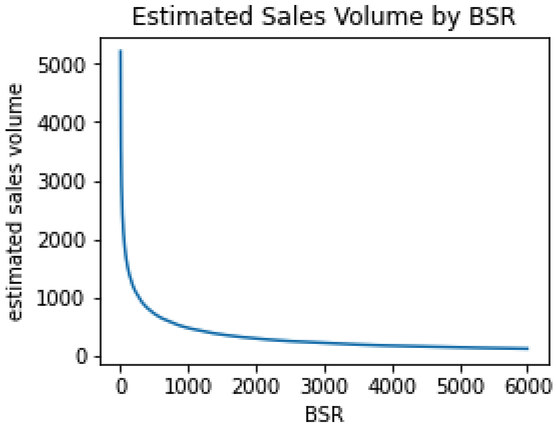

Figure 2: Estimated Sales Volume by BSR

3.3. Reviews

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 3 presents a 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.

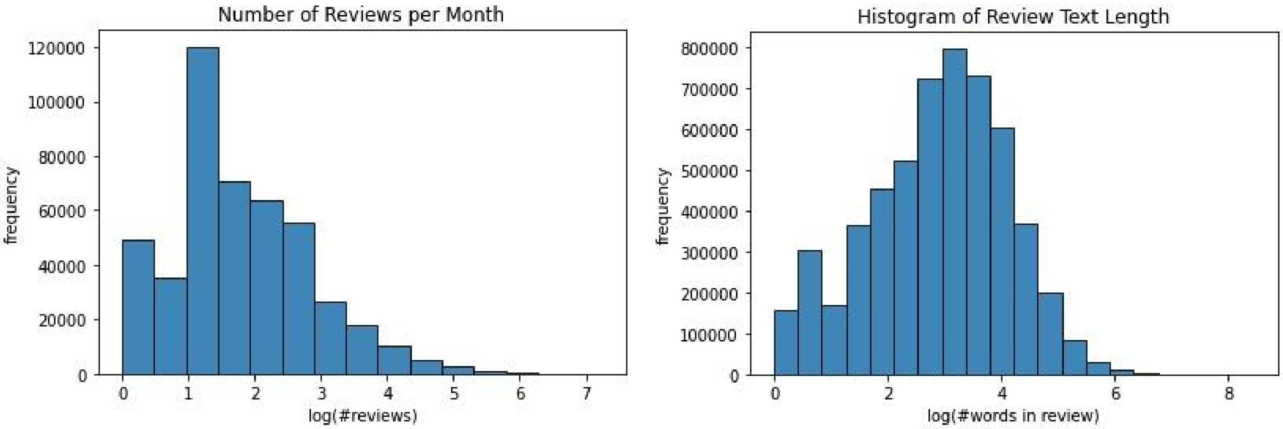

Figure 3: Histograms of the number of reviews and length of reviews

Among all review metadata, review rating is one of the most salient features that influence customer decision. Figure 4 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.

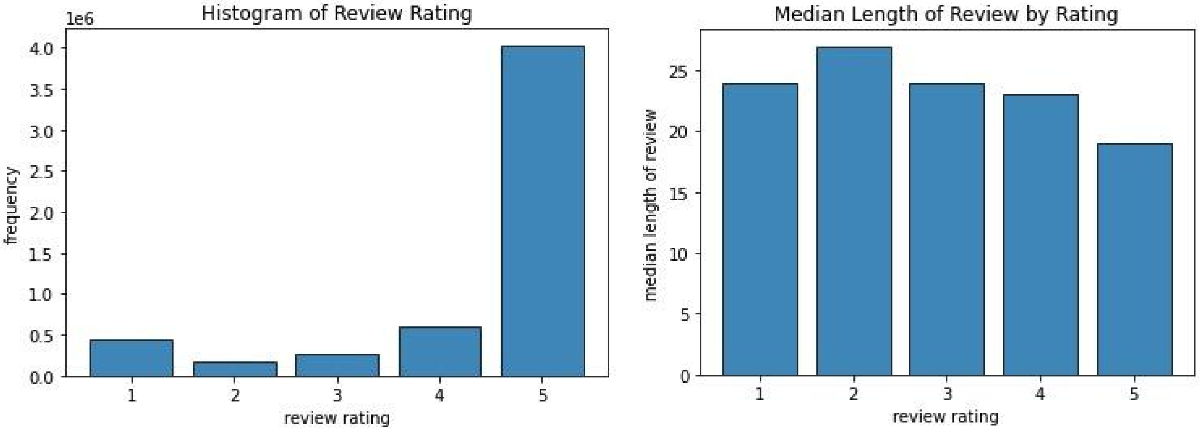

Figure 4: Frequency and review length by review rating

Lastly, we examine the number of upvotes corresponding to each review. Reviews with different numbers of upvotes 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 may notice and trust this review. Figure 5 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 review is 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 upvotes are a snapshot as of the time of scraping, i.e. the number of upvotes represent the number of people who find the review useful between the time when the review is posted and when the reviews were scrapped.[2] We do not know, and have no way of backtracking, the exact timing of the upvotes. As a result, directly using these upvote numbers is prediction essentially entails using future popularity of a review to predict past sales performance. For instance, using the number of upvotes as of January 2022 to predict the sales volumes in January 2021 is inappropriate, as the same review might be less popular in January 2021, and, in turn, has a lesser impact on consumer decisions.

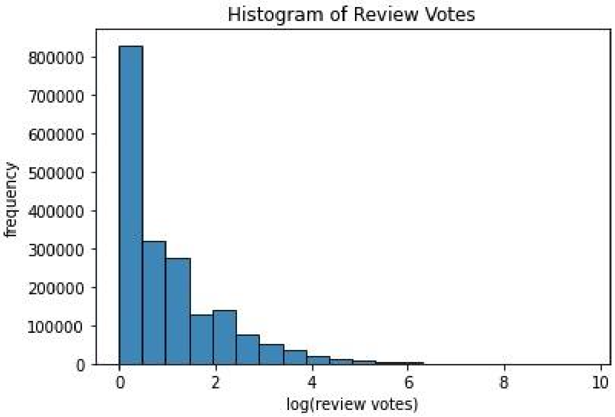

Figure 5: Histogram of the number of review upvotes

[2] Roughly in January 2022.

4. Data Processing

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

1. We exclude products with more than 30 days of consecutive missing BSRs. For the remaining products, we fill in the missing values by taking the median of the BSR right before and after the missing range.
2. To reduce the noise in the data, we smooth the BSR data by calculating a 10-day rolling median. We refer to this rolling median BSR as “rolling BSR” for ease of reference in the rest of the report.
3. We then group the data by product-month and calculate the product’s median BSR for each month. We use the (lagged) monthly median BSR as the target variable in some of our models. We refer to this quantity as “*monthly median BSR*” throughout the report.
4. Lastly, we normalize the monthly BSR and the rolling BSR. In either case, we first compute its maximum and minimum values observed in the dataset, then we use the Max-Min method to scale down the value of BSR to between 0 and 1. In this way, the worst rolling (monthly) BSR is normalized to 1 and the best rolling (monthly) BSR is normalized to 0.
5. Finally, We reshape the data to yield a product-month level dataset.

Note that the rolling BSR has different sizes across months. For instance, there are 28 or 29 days in February, whereas there are 31 days in December. To obtain the same number of rolling BSRs across observations (recall each observation is a product-month), we drop the first rolling BSR in months with 31 days, and add the monthly mean of rolling BSR to the beginning in months with fewer than 30 days.

We take the following steps to pre-process the estimated sales data:

1. For ranks without an estimated sales volume (i.e. ranks larger than 454,302), we extrapolate the estimate using a linear line between 0.005 (which is the estimated sales volume for rank 454,302) and 0.
2. Then we merge the estimated sales data with the original BSR data using rank, to give each rank a corresponding estimated sales volume.
3. We then group the data by product-month and calculate the product’s median sales volume for each month. We use the (lagged) monthly median sales volume as the target variable in some of our models. We refer to this quantity as “*monthly median sales*” throughout the report.

Figure 6 and 7 below plot the distribution of the two potential target variables, monthly median BSR and monthly median sales. We note that the distribution of monthly median sales is extremely right skewed—while the largest value is around 5000, the vast majority of observations have monthly median sales below 100. The median observation has a monthly median sales volume around 40.

Lastly, we take the following steps to pre-process the review data:

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

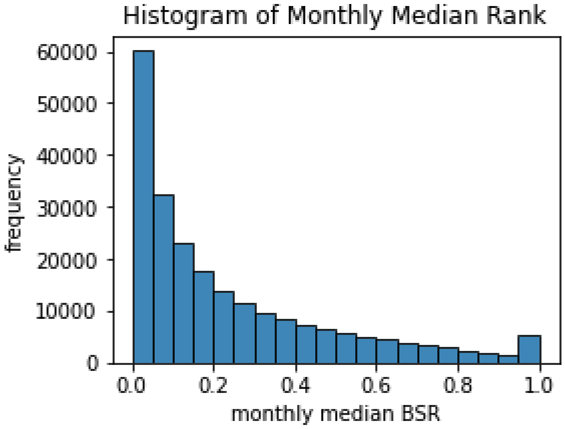

Figure 6: Histogram of monthly median rank

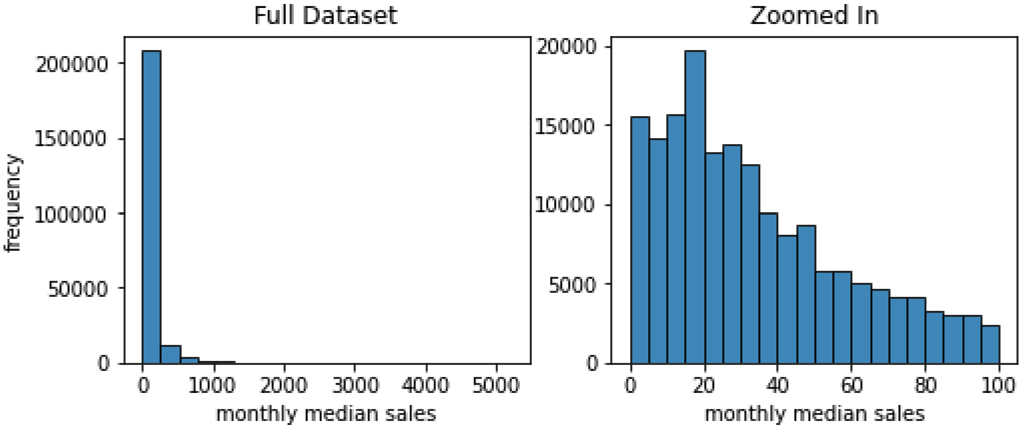

Figure 7: Histogram of monthly median sales volume

5. Literature Review

Recent studies have shown that predicting sales through online reviews is reliable. The study in paper [1] examines the effect of online reviews on new product sales from Amazon.com. The study also found that sales prediction models that include sentiment variables improve the ability to fit the data and the predictive power of the model. In [2], the authors use large-scale text mining to characterize the behavior of e-commerce consumers and model the relationship between product presentations and business outcomes. As described in [3], the authors establish 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 [4], the authors present a sentiment-aware model for predicting sales performance using blogs.

For sales prediction, some commonly used models are linear regression and tree-based models like XGBoost [5]. Regarding performance evaluation metrics, [6] used R2 and RMSE to evaluate the performance of regression models by comparing the predicted sales with the ground truth value. For products review text analysis, Transformers and its variants are proved to be reliable, as described in [7]. Transformers, proposed in [8], are widely used for text encoding. A transformer is an encoder-decoder-based neural network that aims to solve sequence-to-sequence tasks while handling long-range dependencies with ease. Its main feature is the use of so-called self-attention (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) to compute representations of sequential data, such as natural language text, without needing to process the data chronologically. BERT is a state-of-the-art language model proposed by researchers at Google AI language in [9]. It is a transformer-based technique for learning representations of language and is widely used for text representation as well. More specifically, it is applying bidirectional training of Transformer to language modeling, which achieves a deeper sense of language context and flows than a single-direction Transformer model.

6. Models

We have developed two different types of regression models

1. models without any review text in the input which only uses review metadata, and
2. models based 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. This helps 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

6.1.1 The models and the results

Since Milestone 1, we have completed all of our previously mentioned goals for the regression models without text. We have also changed several aspects of our models in terms of features and target variable considered, that we present below.

Our regression models without text consist of linear regression, XGBoost, and random forest models to predict the future estimated sales of a product based on most of the data available apart from the text of the reviews and their titles. For every product, we predict the next monthly median sales using the following features:

1. The rolling BSR (this constitutes 30 values).
2. The monthly median BSR.
3. The monthly mean review ratings.
4. The monthly mean sales.
5. The monthly median sales.
6. The monthly mean price.
7. The monthly median price.
8. The monthly mean 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.
9. The average over all the monthly review ratings mean of all the previous months, i.e. the average over the values of point 3 calculated for every month.
10. 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 8 calculated for every month.
11. The cumulative number of reviews over the previous months.

In total that represents 40 features. The last three features are cumulative data 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.

In comparison to the previous models developed for Milestone 1, we have added the features related to sales volume, as we have decided to change the target variable from change in rank to monthly median sales. This decision was made because a good product which stays good or a bad product which stays bad would both show very little change in rank. It is thus hard for the models to differentiate between those two cases based on review data. In other words, the reviews for a constantly bad product or constantly good product are very different but both have very little change in rank. We have also added the information on the price which could greatly influence the decision of a customer to buy a given product. The cumulative number of reviews was also added as a feature since Pattern was interested in understanding its predictive power. Finally, the features related to upvotes (or the count of people who found a review useful) were dropped as they were a form of data leakage. Indeed, as the numbers of upvotes were only recorded when the data set was created, they were not informative of the importance of a review at a given month in the past. It could have been for example that a review got almost no upvotes at a given month and received a lot of upvotes only later just before the data was collected.

We have then inputted these features into linear regression, XGBoost, and random forest models imported from scikit-Learn. These models were then trained on a third of the products and tested on another third of the products[3] using the computing resources of AWS. After optimization the hyperparameters of our models, the resulting performances were found to be:

Models	**Hyperparameter**	**R-squared score**
Linear regression	———————————	0.938
Xgboost	learning rate = 0.05 n estimators = 100	0.946
Random forest	max depth = leaves pure[4] n estimators = 500	0.939

The R-squared scores are extremely high for these very simple models. It was first hypothesized that this was due to some problem in our data. Indeed, we thought that the estimated sales volume of most of the products we were feeding the model and/or testing on were very close to zero. This would result in a trivial prediction for most of our test data-points, which would result in a high score. However, after inspection of our data, it became clear that this was not the case as can be seen in the histograms of the distribution of the data in figure 7.

The reason for the very high scores became clearer with the analysis of our models which we present below.

[3] We have decided to restrict ourselves to a third of the products both for training such that we could train the models more rapidly. This is especially important for the text-based models which take much longer to train and as we wanted to compare performances with these simpler models, we had to train and test them on the same data.

[4] This is the default setting, i.e. if max depth is None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

6.1.2 Analysis of the models

We first explored the importance of our features using permutation feature importance. The following histograms present the importance of the five most relevant features for both the Xgboost and the random forest models respectively.

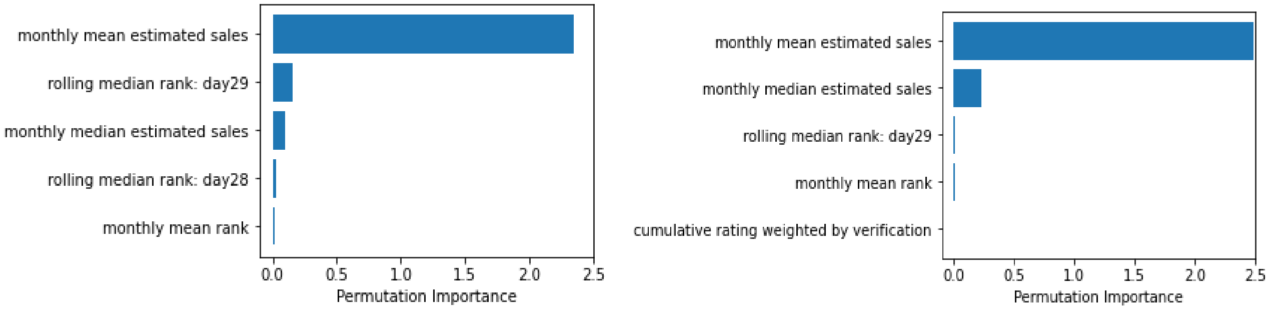

Figure 8: Histograms of feature importance (left: Xgboost, right: random forest)

We see that both our models are strongly dependent on a single feature, namely the monthly mean sales. It is thus not so surprising that the models were having very high R^2 scores. Indeed, it is intuitively clear and expected that the models will be very good at predicting the next month's sales volume given the current month's sales volume as the predictions are based on the principle of momentum, i.e. a product which is selling well this month will probably continue to sell well in the near future. It is however unclear why the models are both basing the next month's median sales prediction based on the current month's mean sales. One would expect that the models would use the current month's median sales. This is probably due to the fact that the monthly median sales and monthly mean sales are highly correlated and the models picked up the monthly mean sales instead of the monthly median sales as means contain more information about the data than medians.

We have then considered partial dependence plots to analyze the dependency of the target variable on certain features. We discovered that the most relevant feature, i.e. the monthly mean sales, had a linear relationship with the target variable for both models, as can be seen in the following plots.

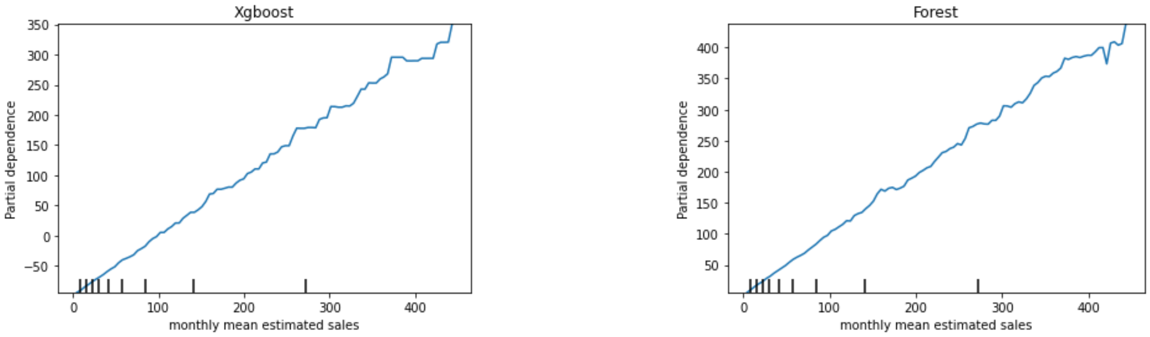

Figure 9: Partial dependence plots for the most relevant feature for both models

This linear relationship supports the intuition that predictions are based on momentum. We also computed the partial dependence of the second and third most important features of both models. The second and third most important feature for the Xgboost model and the random forest model respectively is the rolling median rank on the 29th day of the month. The resulting partial dependence plots also have the expected behavior as a small rank leads to high predicted monthly median sales for the following month (see figure 10). The third and second most important feature for the Xgboost model and the random forest model respectively is the monthly mean rank. In this case as well we observe the same expected behavior, i.e. a small rank leads to a high median estimated sales volume predicted[5]. We note that both of these features are correlated with the monthly mean estimated sales as we have a one-to-one function translating estimated sales volume to BSR and vice-versa.

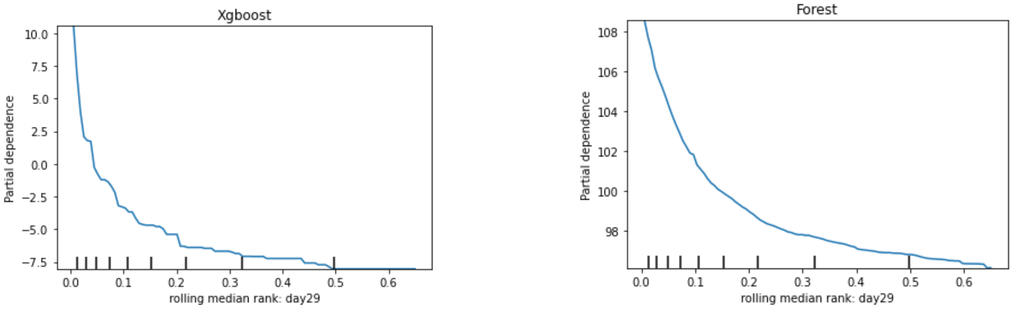

Figure 10: Partial dependence plots for the rolling median rank on the 29th day of the month.

We also considered the partial dependence plots of other features of interest such as the cumulative number of reviews, but they are hard to interpret as they are not relevant enough for the predictive power of our models.

6.2 Regression with Text

Compared to Milestone 1, we have now added a set of bag of words based models, and have made multiple iterations on the transformer models.

6.2.1 Bag of Words Models with Linear Regression

We use a bag of words model as the benchmark for prediction using only reviews text. Such a benchmark model has two major merits. First, it is easily trainable and provides a baseline performance that can be compared with a more complex transformer based model. Second, a bag of words model is highly interpretable. For instance, we can simply look at the phrases that are associated with positive or negative coefficients in a linear regression, and gain some insights on what topics are associated with high or low sales performance.

The target variable in this case is also the monthly median sales of the next month, identical to that in the non-text model. The predictors are the (weighted) frequency of the 500 most common phrases in the training corpus. For text processing, we experiment with the bag of words model, where we simply count the occurence of each of these 500 phrases, as well as the TF-IDF model, where the weighted frequency of each phrase is calculated from the term frequency and inverse document frequency. The TF-IDF model has the advantage of adjusting for the fact that some words are used more commonly in general. We then use a regularized linear regression model to predict the monthly median sales of the following month.

The set of hyper-parameters we experiment with includes: simple bag of words model vs. TF-IDF model, bigrams vs. trigrams, LASSO vs. ridge regression, and the different penalty strengths. Figure 11 tabulates the R^2 score on a hold-out validation set across different

[5] As the partial dependence plots for this feature are very similar to the rolling median rank on the 29th day of the month we omit the figure and refer the reader to figure 10 as a reference.

model/hyper-parameter combinations. With the bag of words model, the R^2 score is sensitive to changes in hyper-parameters; however, the R^2 score is more stable with the TF-IDF model. The best performance is attained with a bigram TF-IDF model with a LASSO regression using a penalty strength of 0.1. The R^2 score is quite high, especially given that we are only using a simple linear model on a very sparse model matrix.

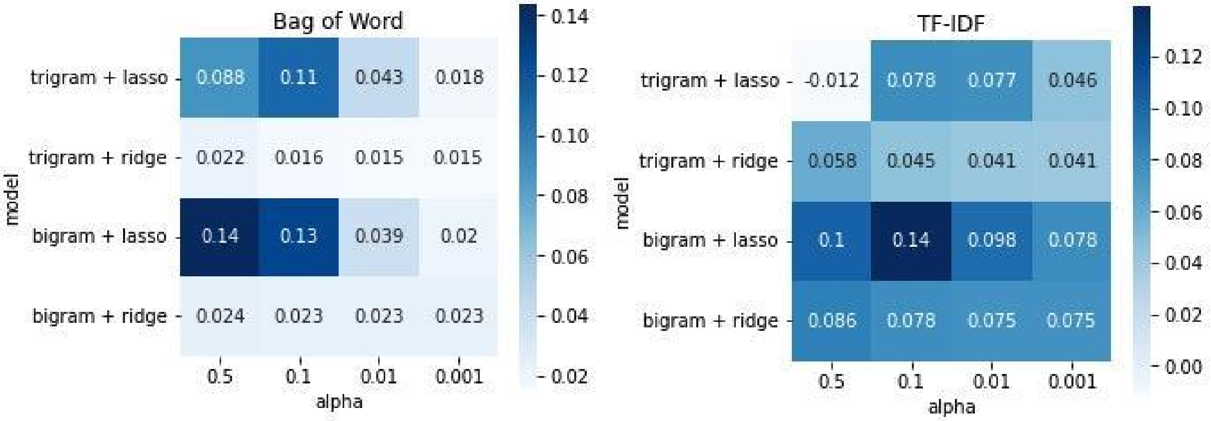

Figure 11: R^2 score across different model/hyper-parameter combinations

To better understand the model predictions, we show a binscatter plot of the target value against the model prediction in Figure 12. The binscatter plot is generated as follows: we divide all observations into 30 bins based on their model prediction values. Each bin has the same number of observations. Then we plot the average true target value against the average prediction value within each bin. Intuitively, the closer the points are to the 45 degree line, the better the prediction is compared to the real values of the target.

We see that the predictions below 200 are unbiased—the predictions match the corresponding target values closely. However, the predictions above 200 are biased downwards—the average true value is around 400 when the model predicts an average of less than 300. This pattern is not surprising given that a linear model can hardly capture the nonlinear pattern of the monthly median sales of the very successful products.

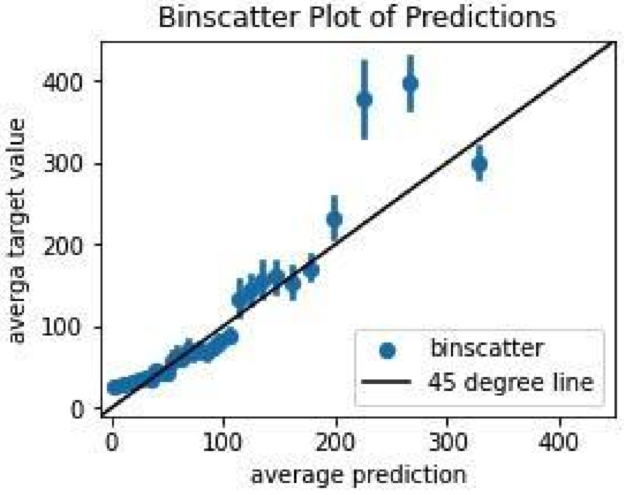

Figure 12: Binscatter Plot of Target Variable vs. BoW Predictions

Focusing on the best performing model, we further examine the bigrams that are the most predictive of sales volumes to gain some insights on influential keywords and topics.

The 10 bigrams associated with the largest positive coefficients are as follows: *cider vinegar, garden life, bowel movements, flu season, hair nails, taking probiotic, 2nd bottle, taste great, getting sick,* and *stopped taking*. The only 2 bigrams associated with negative coefficients are *joint pain* and *pain relief*.

Many of these bigrams are quite intuitive although some require more context that is not captured by a bigram model. We make two observations. First, “garden of life” is a famous brand name of vitamin products, and is one of the best sellers on Amazon. The model precisely picks up the brand[6] from the review texts. Second, “2nd bottle” is predictive of high sales performance. There may be two ways through which this keyword is associated with high sales volumes. First, it is the consequence of a repeated purchase, and therefore represents the increase of the intensive margin—the product is retaining existing buyers. Second, it increases the extensive margin—new customers may be convinced by such a genuine review and start to purchase this product.

6.2.2 Transformer Models

Since Milestone 1, we have added many iterations to our transformer model. This section will be structured by listing out each version of the model as a separate section, and then discussing their results in the final section.

6.2.2.1 Transformer Model Version 2

There were 3 major changes to the model described in Milestone 1, that led to this version of the model:

1. We first had to get rid of the upvotes as a feature, which were previously playing a central role in the model - both as input and as a weighting factor to create the embedding of the aggregate review information. This was required because, after discussions with Pattern, we realized that upvotes were a form of data leakage - i.e. the data was collected at the end of the timeline for each review, and so we didn’t know the upvote values for different points in the timeline.
2. We wanted to use a simpler transformer model - specifically move from large BERT to tiny BERT. This was because the gain in performance for large BERT over tiny BERT wasn’t very significant, and the training time for large BERT made prototyping of the model close to impossible because of the long training time.
3. We also trained our model for a significantly larger number of train steps (going from 200 steps previously to 50,000 now). We were able to make this change because we now had access to AWS, and could run experiments remotely. This change is very significant, but even then only regresses on a training point once.

A diagram for the new model is shown in Figure 13 below. It is important to note that the aggregate review embedding represented here is now no longer weighted by upvotes, but is instead a simple mean of all the review embeddings.

6.2.2.1 Transformer Model Version 3

After having trained the previous model, we realized we needed two more changes to the above described structure:

[6] In pre-processing, we take out all stopping words in the English dictionary (e.g. an, the, of). Therefore, “garden of life” is pre-processed to be “garden life.”

1. We had to get rid of the previous BSR input because as we saw from the non-text analysis, any information about product revenue momentum was extremely predictive of the target, and as such we knew that our model was only going to consider that information if we passed it into the model. Instead, we needed a way to create a fair comparison between our transformer model and the bag of words model described above. We also wanted to answer the question - how much of the variance in the target can be explained by review text alone?
2. We also wanted to change the target variable from the BSR change to the next month's sales volume. This, again, was done to make the transformer models more inline with the other models we had.

The final structure of this model is described in Figure 14.

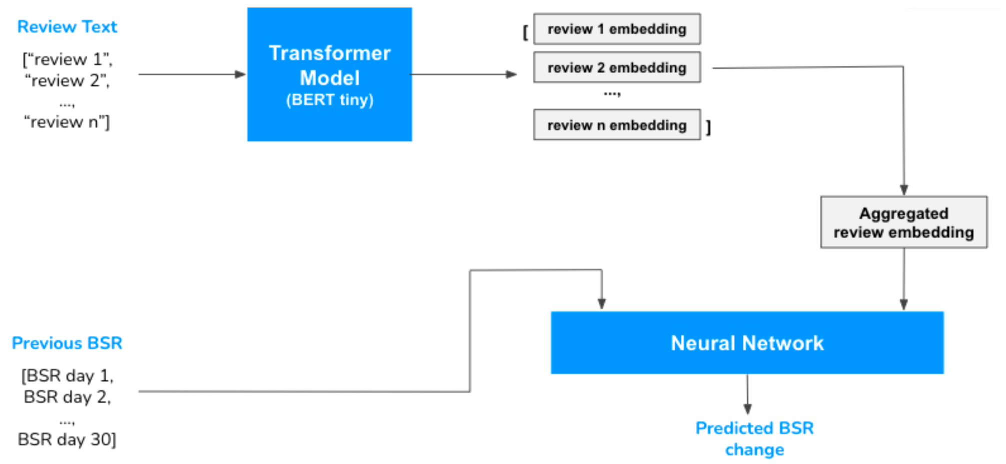

Figure 13: Transformer Model version 2

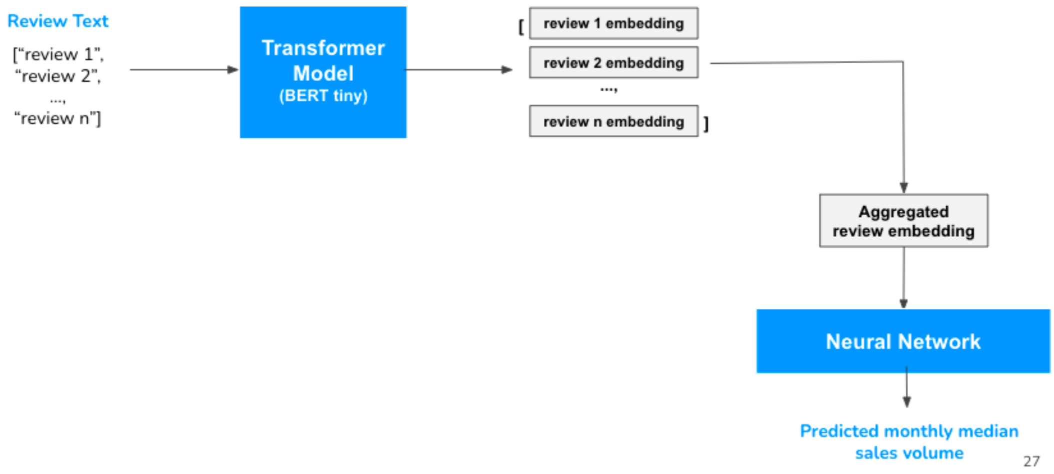

Figure 14: Transformer Model version 3

6.2.2.3 Transformer Model Results

The results (R^2 score) for the models is shown in the table below. Please note that the test set used for this table is exactly the same test set that is used for the other models in our report (namely the non text model and the bag of words model).

Transformer Model Version	Total Train Steps	R^2 Score
Version 1 (Milestone 1)	200	-0.132
Version 2	50,000	0.256
Version 3	50,000	0.164

6.2.2.3 Transformer Model Discussions

In this section, we give reasons why we believe the above R^2 scores were seen for each of the models described.

For the first model, from Milestone 1, the reason for the bad R^2 score is clear - the model was severely underfitting the data because it had only seen about 0.4% of the training data, which, of course, doesn't even begin to capture the variance of the target variable. However, the results from this model were included here to illustrate a point about how important the training steps are in improving the model performance. This might seem an obvious point to make, but it is one of the major points we will make for our discussion of model version 3.

For the version 2 model, which was the best performing model out of transformer model, we see the immediate impact of adding the train steps into the training process - even though version 2 lacked an important feature that version 1 had access to (review upvotes), it was able to capture up to 25% of the variance of the target variable. The reason for this high score is potentially because of the inclusion of the previous BSR values in the model. As we saw in the non-text analysis, when the time window is as short as 1 month, momentum information is the most important feature for predicting the next month's target. This leads us to believe that this transformer model structure was not getting much information gain out of the reviews themselves, but was instead using the previous rank values almost entirely to make the prediction.

For version 3 of the model, there is a drop in the score compared to version 2, but this is not a fair comparison to make for two very significant reasons: (1) they are both predicting on different targets, and (2) we have dropped the most important feature for version 2 from our version 3 model. However, we can make a fair comparison of this model with the bag of words models described previously. Specifically, the best performing bag of words model had an R^2 score of 0.138, while the transformer model has a score of 0.164. This, at first, did not seem like a significant improvement to us, but after discussing our results with Pattern, they mentioned that the results were what they expected to begin with. This is because the increase is a 10% increase in performance, and that is what is expected for most regression based tasks that do not have a very significant context based component compared to the vocabulary based component.

Nevertheless, we believe there are two ways this model can be improved, both of which are from the assumption that the model is currently underfitting the train data. The first of these involves looking at the training steps. Although the model was trained for 50,000 steps, this is not a large number compared to the training data, and it means the model weights have regressed on each training point just once during the training cycle - multiple such updates on each training point will allow the weights to be shaped even better for the train fit. This conclusion of letting the model train more is also backed by evidence from the 5 train epochs themselves, which show the following trend in R^2 score: -0.213, -0.003, 0.018, 0.115, 0.139. It is clear from this trend that the R^2 score has only been increasing, and stopping at 5 epochs was a form of regularization because it was early stopping, and there was more potential for the R^2 to increase. The big challenge here is that the model currently takes 3 days to train, which leads to much longer prototyping cycles.

The second improvement we can make is related to the model structure- specifically, the number of trainable parameters. Currently, our model uses vector representations for reviews that are 40 elements in dimension. So, when we compare this to the 500 element vector that the bag of words model is making a prediction based on, we see a big difference in the amount of information the model has to make a prediction. To improve this, we can simply increase the size of our review descriptors, while also adding more dense layers in our neural networks, so that we can better capture the non-linear trends that we are probably missing out on right now.

In Figure 15 below, we see a binscater plot for our version 3 model. This plot helps us make more sense of the model predictions. To understand this plot, recall that the closer the points are to the 45 degree line, the better the prediction is compared to the real values of the target. On plotting this, and comparing it to the binscatter for the bag of words model, we notice that the transformer model is better able to capture the trends for the tails of the distribution. i.e. it is doing a better job at predicting things for very successful products compared to bag of words. This is what we would expect, since BERT along with the neural network structure has the additional advantages of non-linearity. It is, however, not doing as good a job for the linear trend closer to the 50-100 mark, and this can potentially be explained by the smaller number of train steps this model has had to capture that trend successfully.

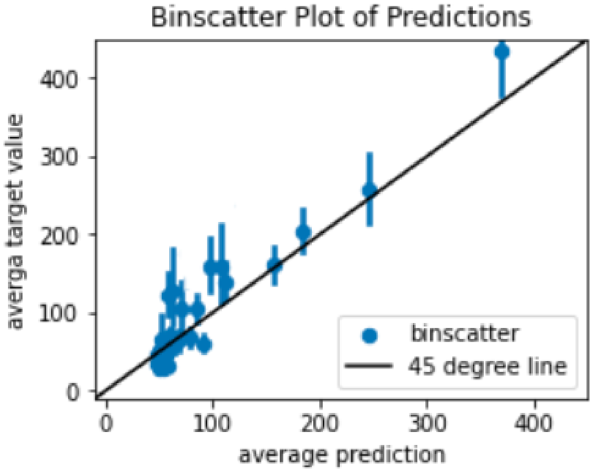

Figure 15: Binscatter Plot for BERT

7. Future Work

Based on our discussion with Pattern during our latest meeting, we plan on changing the prediction task for our models. Indeed, we have discovered so far that our regression models are predicting near-term future performances of a product very well based on the momentum of a product. We would now like to focus on predicting how a product will do in the long term. Indeed, this task would be potentially more useful for Pattern as they could predict for their customers the long term success of their products. The task will now be formulated as follows: based on the review data of the first three months of a product can we predict if it will ever reach a successful rank in the year that follows. A successful rank is a relative measure which we plan on defining as being approximately the top 15% of the ranks recorded for all the products in the vitamins and supplements category. The shift to this new task will be our main focus from now on. We hope that the reviews’ metadata will have a greater impact on the prediction of long-term success of a product and we hope to find some key insights in the text of the reviews for a product to predict its long-term success.

References:

[1] Geng Cui, Hon-Kwong Lui & Xiaoning Guo (2012) The Effect of Online Consumer Reviews on New Product Sales, International Journal of Electronic Commerce, 17:1, 39-58, DOI: 10.2753/JEC1086-4415170102

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

[3] 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.

[4] 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

[5] Chern, CC., Wei, CP., Shen, FY. et al. A sales forecasting model for consumer products based on the influence of online word-of-mouth. Inf Syst E-Bus Manage 13, 445–473 (2015). https://doi.org/10.1007/s10257-014-0265-0

[6] Elizabeth Fernandes, Sérgio Moro, Paulo Cortez, Fernando Batista, Ricardo Ribeiro, A data-driven approach to measure restaurant performance by combining online reviews with historical sales data, International Journal of Hospitality Management, Volume 94, 2021, 102830, ISSN 0278-4319, https://doi.org/10.1016/j.ijhm.2020.102830.

[7] Balakrishnan, V., Shi, Z., Law, C.L. et al. A deep learning approach in predicting products’ sentiment ratings: a comparative analysis. J Supercomput 78, 7206–7226 (2022). https://doi.org/10.1007/s11227-021-04169-6

[8] 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.

[9] Devlin, Jacob, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” ArXiv abs/1810.04805 (2019): n. pag.