**FAIR: Feedback Analysis and Intelligent Rating**

Eliminating non-informative score ratings on shopping platforms once and for all

powered by BART and multi-layer classifier

Baiyi Liao, NYU Class of 2024

**MOTIVATION**

After writing a long review on shopping platforms describing how you like and dislike the product you just purchased, how many starts, ranging from 1 to 5, should you award to the product? If it’s a genuinely good product, what is the difference between 4 stars and 5 stars? If you do not have a good time with it, should you award 1, 2, or 3 stars?

Unfortunately, this is indeed a hard question for every reviewer, and has led to polarized ratings on shopping platforms such as Amazon: Because people cannot objectively quantify their feelings to a numbered scale, many customers will treat the star rating section as a binary toggle – 1 star if they do not like it, 5 stars if they like it. Because of this, most researchers attempting to implement rating predictions build binary classifiers to tell if one review is simply positive or negative, which, when applied to industrial cases computes a continuous average score from it – usually 1 for positive and 0 for negative, will be prone to the imbalance of input datasets of different products due to the difference in initial exposure to the market and the number of pre-existing reviews.

Therefore, this kind of rating is not very informative when we compare two products of the same category using this metric by looking at the average of the star ratings in two decimal places. By the same token, the product recommendation systems of said platforms cannot rely on the star ratings but have to retreat back to sponsorships and biased reviews from “professionals” – influencers who receive products for free. In the end, such systems usually take a lot of labor and hidden costs but yield unreliable results.

Our FAIR system, or Feedback Analysis and Intelligent Rating, aims to tackle this issue by extracting key qualifications of products from the review texts using a BART-based Zero-Shot Sequence Classifier trained on the MultiNLI dataset (Transformers, <https://arxiv.org/abs/1909.00161>), and then build prediction model with a multi-layered neural network. Thanks to this powerful classifier, we will be able to accurately capture the emotions and key qualifications even without the keywords explicitly present in the texts. Additionally, since the MultiNLI dataset used to train this model contains texts of different genres, the generalization performance of our finished product may be surprisingly good.

The outcome of this system will be fruitful: Since the model effectively represents a person who has seen, commented, and rated reviews on behalf of everyone in the dataset, it can objectively yield a rating of a product based on review texts, significantly improving the reliability of average scores, enabling customers to compare products with the computed ratings and allowing recommendation algorithms to finally adopt it as a key metric.

**RELATED WORKS**

1. Zero-Shot Classifier

<https://arxiv.org/abs/1909.00161>

1. The BART Zero-shot classifier trained on the MultiNLI dataset we will be using:

<https://huggingface.co/facebook/bart-large-mnli>

1. As previously mentioned, there were customer sentiment analysis that results in binary outcomes, which is not useful for computing a continuous average score for side-by-side comparison and building recommendation systems. The results may be informative if and only if the reviews for individual products are overwhelmingly negative or positive, but this is generally not the case for most products:

<https://www.kaggle.com/code/miroslavgeorgiev/sentimentanalysis>

1. Although it is true that there were some people attempted to implement solutions just like FAIR, there are lots of room to improve:

<https://www.kaggle.com/code/asemsaber/amazon-reviews-sentiment/notebook>

Although this model achieves somewhat high accuracy, precision, and recall, the use of LSTM and 9-million-parameter embedding model in the first layer may hinder its generalization performance as well as working with edge cases such as sarcasm or passive aggressive. By contrast, the BART architecture is context-aware just like LSTM but better because it’s effectively a hybrid of BERT and GPT, and the trained model used by us has as much as 407 million parameters, which suggests that it would be able to comprehend texts and capture key ideas much more accurately and reliably.

1. There were academic papers describing a pipeline with advanced models to predict the polarity of the comments written and detect falsely rated comment with a Web Interface:

<https://arxiv.org/pdf/1904.04096.pdf>

This paper seems to have a complicated process to predict scores, but I doubt that the predictions can help tackle the polarized reviews problem mentioned earlier as the input dataset itself is skewed to five stars just like the dataset we are using. To solve this issue, we will attempt to sample evenly across the rating classes. From here, we should be able to get a model that not only determines whether one comment is positive or negative that they have done to their web interface, which means that they are not confident with their estimates, we should be able to predict meaningful scores that we can apply directly.

**METHODOLOGY**

Since we are building a model that produces reliable results, we need to filter out the reviews that are not informative in the first place. In this case, a naïve way, assuming that most customers are not intentionally using ChatGPT to write and lengthening their response since there’s virtually no incentive to, to tell if one review is informative is by simply looking at its length. Depending on how the data is distributed in terms of length, we will probably drop any reviews that has less than k words where k is some fixed number between 1 to 10.

Then, we will supply the texts and possible key ideas (such as “good”, “bad”, “high-quality”, “broken”, “scratch”, “fainted”, “new”, “careful”) directly to the Zero-Shot Classifier. Unlike most implementations, we do not need to remove the punctuations and stop words from the raw texts because they have little to no contribution to emotion or would affect text comprehension because our bart-large-mnli model would most likely ignore meaningless tokens when it comprehends text and extract ideas. With that said, if we find something that will throw off our algorithm one way or another, we will remove them.

Now, we should create a keywords data frame where we have indexes of each comment in the first column, and we have shared keywords with weights computed by the bart-large-mnli model which ranges from 0 to 1 continuously, and lastly the true rating from the original dataset.

Next, we split this keywords data frame into a train and test set with 80% in the train set and 20% in the test set. After which point, we will train a multi-class classifier containing multiple layers to aim for at least 85% accuracy. To find out the best parameters, we will use k-fold cross validation for this step. The output should be in reviews ranging from 1 to 5 but in integers since we are treating the rating as a categorical variable.

Finally, when unseen data comes in, we convert the raw reviews into the keywords-with-weights row with the bart-large-mnli model and then predict a rating for it. This would conclude the project.

**DATASET**

<https://www.kaggle.com/datasets/abdallahwagih/amazon-reviews/data>

This is a dataset contains reviews of the Cell Phones & Accessories section on Amazon. It has about 200,000 entries with review text, star, review time and more for each entry. Although this cannot represent all of Amazon since it’s only data from one product category, we believe that the model can be generalized to other categories and shopping platforms as well since there is no evidence that shopping for electronics online does not apply to some subsets of the population.

**WORK PLAN**

Since I am the only person working on this project, the work plan would not include who is in charge of what but rather what needs to be done each week.

**Week 1** (Current Week)**:** Research, Data Collection, and proposal write-up. This is already done since we now have a clear path to move forward with the project: data is found, pre-treating model is also found.

**Week 2:** Data Synthesis and BART model test. We want to see if BART performs well enough for this use case. Although the Zero-Shot Classification paper claims that this paradigm even considers durability when used against unseen data, we need to make sure. After validating the effectiveness, we would find the template words the classifier uses to generate a [0,1] score for each of the words in the dataset. Since this is a GPU-heavy task, I have to dedicate one week for this step.

**Week 3:** Design and train the multi-layer model. This step is straightforward: play with a model that has few layers and finetune it with k-fold cross validation. If we can identify the optimal loss function, activation function for each layer, number of epochs, and batch size, we should be done for the week.

**Week 4:** Write a server that accepts user queries for making predictions. This will then allow us to do a demo app. With the demo app, it will be possible to then have a proper final paper, a presentation, and a cleaned-up GitHub Repository.