Prediction of Responses To Binary Customer Questions With Answer Data

Cyril Gorlla

Prediction of Responses To Binary Customer Questions With Answer Data

Cyril Gorlla
cyril.m.gorlla@jacobs.ucsd.edu
University of California San Diego
La Jolla, California, USA

```
is it user friendly
can i play facebook games on it
do you carry this product in white ?
would this product be useful on a iPod touch 64 GB. Model # MC011LL
can you download/save apps in this card?
is mini hdmi same as micro hdmi
is this work with canon rebel t3?
can I trim back plastic in mini side a bit?  I have a mini micro port that is a bit deeper.
are there different sizes of Mini HDMI
do u need to already have phone service for this to work
is this cord used to connect mybtablet to my flat screen tv so i can watch the internet on
```

Figure 1. Example of binary questions asked on Amazon.

Abstract

Within the context of online shopping, it is often imperative to respond to customer queries promptly. In this paper, we explore and evaluate various methods of responding to binary customer questions with machine learning algorithms. Specifically, we implement models based on logistic probabilities as well as collaborative filtering in order to predict the response to binary questions on Amazon. We find that probabilistic models fare the best on our validation set (which includes unseen data), with the logistic model trained with vector embeddings achieving 67.6% accuracy.

CCS Concepts: • **Computing methodologies** → **Machine learning**; **Artificial intelligence**.

Keywords: neural networks, natural language processing, machine learning, artificial intelligence

1 Dataset

The proliferation of online shopping has naturally led to a large amount of questions being asked by customers on various websites, the most notable being Amazon. Here, we utilize a subset of a dataset[6][10] consisting of Amazon products, along with questions and answers related to the products.

We selected a subset of the dataset that concerned electronics, amounting to 231,449 questions and 867,921 answers. The dataset contains information on the product ID (ASIN), the questions asked about that product, and answers for each question. A feature denoting whether the question is binary or open-ended also exists, as well as polarity scores for answers in the former category. We then refined the dataset to only include binary questions, defined as being able to be answered by a "yes" or "no".

After this, we were left with 61,222 samples. First, we explored the breakdown of the answers (Fig. 2). As we have a $2:1$ imbalance of "yes" to "no" answers, we may either represent this as an accurate distribution of answers in the real world, or account for the differences in building our model. Next, we looked at the length of the questions (Fig. 3). The distributions of length were fairly identical. Finally, we examined the polarity score (Fig. 4). As the binary label was based on the polarity scores, it is sensible that the "yes" and "no" scores are respectively very high and low.

Table 1. Length Statistics

Mean	50.198
Std. Dev.	38.56
25%	29
75%	59

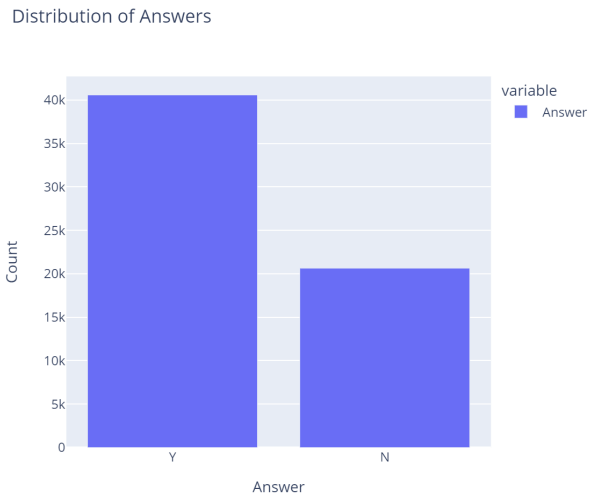

Figure 2. Answer Distribution

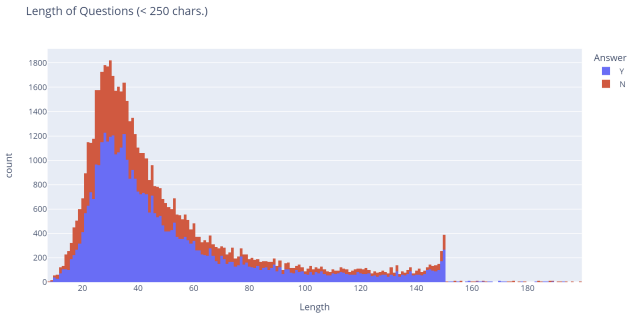

Figure 3. Question Length

2 Predictive Task

Having now filtered the dataset to include only binary questions and binary responses to those questions, we proceed with formalizing the predictive task. Given some features for a binary question $i : \{i_1 ... i_n\}$, we wish to predict the answer to this question in $\{Y|N\}$. Features that are relevant for use in such a task consist of the product that the question is being asked about, as well as features pertaining to the question itself, such as the length and text of the question. It is also necessary to determine how to generate a proper label for each question. We use the most frequent binary label that appears in a question's answers to determine the "ground truth" label for that question. In cases of a tie, the label is randomly selected. This occurs in .06% of the data.

As the data was "nested", that is, each product contained a list of questions, and each question contained a list of answers, it was necessary to "flatten" the data such that we had a single row that contained the product ID, a binary question about that product, and the labelled response for that question. We also assigned a unique number to the products

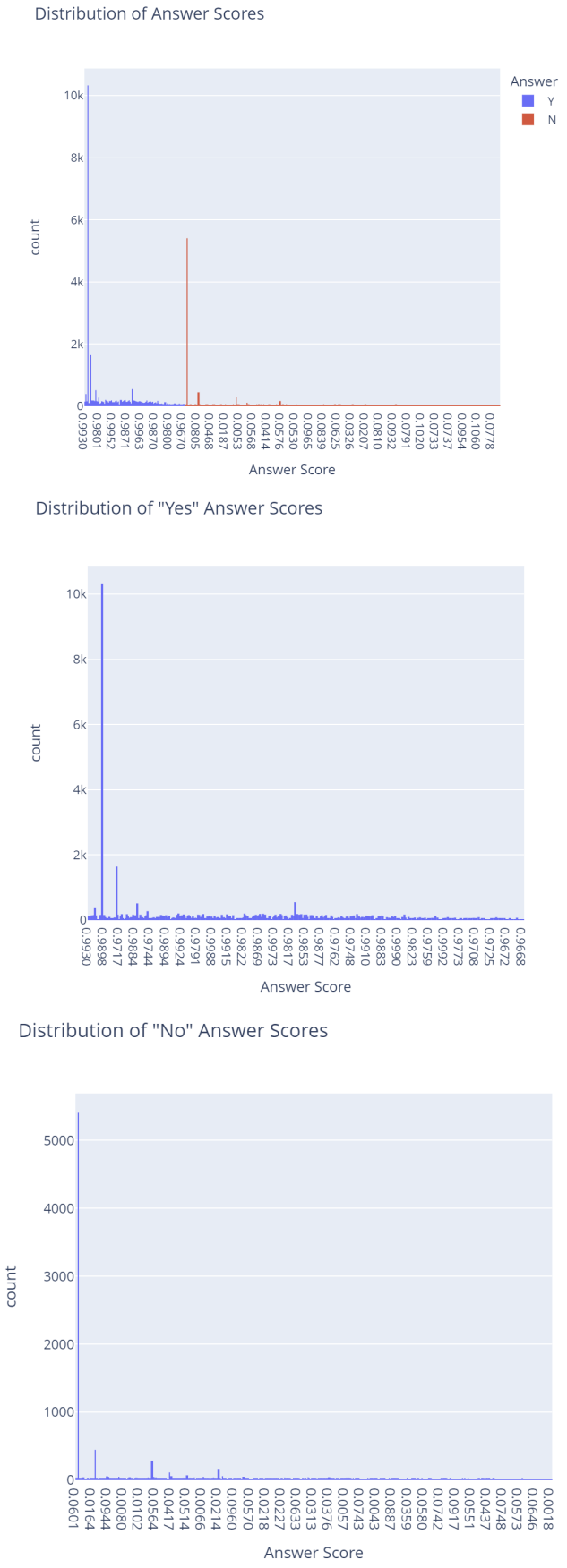

Figure 4. Polarity Distributions

and questions in order to create valid training vectors. Word vector embeddings may also be predictive in this task, and so we utilize Word2Vec [7] in order to train two word vector models: one trained on "yes" questions, and the other trained on "no questions". If there are semantic differences within these questions, the higher of the probability scores when inputting a question into the two models may be a valid signal for our model.

Two main models were used: a logistic model and a collaborative filtering neural network model. For the logistic model, we can utilize a standard train-validation split of 75%-25%, and due to the randomized nature of the split, we can utilize accuracy as our evaluation metric. For the neural network model, we utilize the same split but use binary cross-entropy as an evaluation metric to rectify the imbalance. The logistic model allows us to have a simple model with the notion of probability, with the sigmoid function scaling the output from 1 to 0, where 1 is "yes" and 0 is "no". The neural network allows us to utilize patterns of overlap in products and questions.

For a baseline for our model, we can utilize a naive input of the products and questions encoded as integers into the collaborative filtering model, with no additional processing.

3 Model

The final logistic model was trained with scores generated from the Word2Vec models. Word2Vec tries to find [8]

$$R : \text{Words} = \{w_1, ..., w_N\} \rightarrow \text{Vectors} = \{R(w_1), ..., R(w_N)\} \subset \mathbb{R}^d$$

such that:

$$w_i \approx w_j \qquad \text{(meaning of words)}$$

is equivalent to:

$$R(w_i) \approx R(w_j) \qquad \text{(distance of vectors)}$$

If a particular question had a higher score from the "no" model, the feature for that question would be $1 - score$, or just the score if the positive model was higher. Unlike the collaborative filtering model, the logistic model does not capture notions of product-question overlap, but is able to more directly represent patterns of probability in the data. Other features and combinations of features like length were implemented, but did not function as well.

The final collaborative filtering model utilized the length of the question and the product ID to predict a response to the question. Similarly to the logistic model, other combinations of features proved less useful. A batch size of 8 constituted the best performance in training the model. We define the notion of similarity in this model with the dot product of the vectors, that is

$$sim(i, j) = \overrightarrow{i} \cdot \overrightarrow{j} \tag{1}$$

.

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 embedding_28 (Embedding)     multiple                  1308850

 embedding_29 (Embedding)     multiple                  26177

 embedding_30 (Embedding)     multiple                  954550

 embedding_31 (Embedding)     multiple                  19091

=================================================================
Total params: 2,308,668
Trainable params: 2,308,668
Non-trainable params: 0
_________________________________________________________________
```

Figure 5. Collaborative Filtering Model

4 Literature

The work that this dataset originated from [10] [6] dealt with addressing customer queries from available product data. The authors chose to predict responses to questions based on review data using a mixture-of-experts (MoE) model, factoring in product reviews and giving each review a "vote". We use a similar notion, although we only use question data and focus on binary questions. Our process of generating ground truth labels is similar in that we examine the binary responses and choose the mode of them, as the most frequent answer is likely to be the correct response to the query.

Other research [2] has focused more generally on answering binary questions, not in the specific context of online shopping. They concluded that predicting responses to binary questions was a difficult task, with their final model scoring 10% worse than human performance at the same task. This general pattern was also replicated in our work, with prediction accuracy generally being meager compared to human performance. BERT, a large language model, was used in that paper to derive meaning from questions. In comparison, the Word2Vec implementation we use is limited and lightweight, only being trained on the test of the questions with each label. Using a model trained on much more general data like BERT may yield performance gains at the expense of significantly increased model complexity.

A pertinent application of predicting question responses is to facilitate chatbots on websites that assists customers with their queries. When this has been implemented [3], parsing text both from the product page as well as user-generated content like reviews and questions/answers has proven to be useful in accurately answering customer queries. While we only look at the ASIN to provide a unique identifier for each product, incorporating details about the product may be a valuable area of exploration.

5 Results

The best model was the logistic model that incorporated Word2Vec similarity scores, with a validation accuracy of 67.7%.

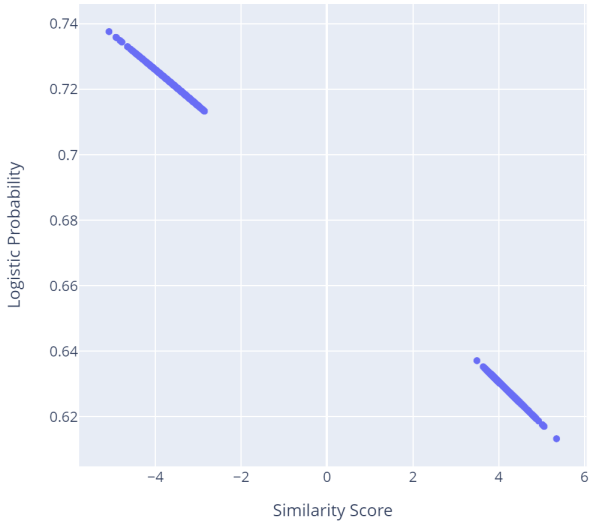

Figure 6. Final Logistic Model Predictions

Our baseline model had a validation loss of .68, which was improved to .65 when replacing the direct question encoding with the question length instead. However, it is apparent that the collaborative filtering model is generally unconvincing when compared to the validation accuracy of the logistic model. This denotes that the product and question vector embeddings are not predictive within the context of the collaborative filtering model. Probabilistic methods, such as our logistic regression model, fared the best on our validation set, indicating that the proportion of "yes" and "no" answers in general is predictive, along with word vector embeddings of the two groups. Further development in this area may find the incorporation of further information about the product and larger language models to be valuable in predicting answers.

References

[1] François Chollet et al. 2015. Keras. https://keras.io.

[2] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. arXiv:1905.10044 [cs.CL]

[3] Lei Cui, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoqun Duan, and Ming Zhou. 2017. SuperAgent: A Customer Service Chatbot for E-commerce Websites. In *ACL (System Demonstrations)*. 97–102. https://doi.org/10.18653/v1/P17-4017

[4] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585, 7825 (Sept. 2020), 357–362. https://doi.org/10.1038/s41586-020-2649-2

[5] Plotly Technologies Inc. 2015. *Collaborative data science*. Montreal, QC. https://plot.ly

[6] Julian McAuley and Alex Yang. 2016. Addressing Complex and Subjective Product-Related Queries with Customer Reviews. In *Proceedings of the 25th International Conference on World Wide Web* (Montréal, Québec, Canada) *(WWW '16)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 625–635. https://doi.org/10.1145/2872427.2883044

[7] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. arXiv:1310.4546 [cs.CL]

[8] Mostafachatillon. 2015. mostafachatillon/word2vec. https://github.com/mostafachatillon/word2vec

[9] The pandas development team. 2020. *pandas-dev/pandas: Pandas*. https://doi.org/10.5281/zenodo.3509134

[10] Mengting Wan and Julian J. McAuley. 2016. Modeling Ambiguity, Subjectivity, and Diverging Viewpoints in Opinion Question Answering Systems. *CoRR* abs/1610.08095 (2016). arXiv:1610.08095 http://arxiv.org/abs/1610.08095

[11] Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman (Eds.). 56 – 61. https://doi.org/10.25080/Majora-92bf1922-00a