

# Instant Feedback on your Amazon Review\*

Cody Wuco<sup>†</sup>  
Computer Science  
Sacramento State University  
Sacramento California,  
Sacramento County  
codywuco@csus.edu

Kevin Estrada  
Computer Science  
Sacramento State University  
Sacramento California,  
Sacramento County  
kevinestrada@csus.edu

Kai Kit Lok  
Computer Science  
Sacramento State University  
Sacramento California,  
Sacramento County  
klok@csus.edu

## ABSTRACT

This project seeks to give instant feedback on a consumer's review. Allowing them to know whether their review will be of value to others or not.

There is an overabundance of reviews on products, and it would be nice to know how useful your review would be when you post it, so you can use the feedback to write better reviews without having to babysit your posts over extended periods of time.

For our approach we will use a dataset from amazon that has reviews that have been collected for 18 years. We will use two models for this project. Convolutional, and Fully connected TF IDf with and without word embedding as well as pretrained GloVe word embedding. We will test the model against reviews and see how well it responds to the review, hopefully providing useful information.

What we will discuss throughout the paper is the problem formulation, what we are trying to address with our inputs and outputs. What the goal of this project is, go through the architecture design of the project, describing the algorithms used in the project. After displaying the results, we will discuss what methodology we used, what the setting of our experiments were, and the different methods we used. Finishing with the results and our conclusion of this whole project.

## PROBLEM FORMULATION

We will use the dataset from amazon on beauty product reviews. Then using the two models mentioned previously. The data will be trained as well as tested, we will use the RMSE score and the lift chart. The Goal is to be able to predict the upvotes based off the reviews left on the product.

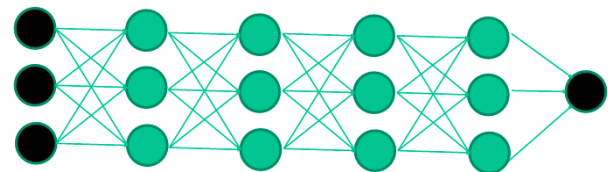
## KEYWORDS

Neural Network, Word Embedding, Tf-idf, Amazon, Reviews, User Feedback, Instant.

## SYSTEM/ALGORITHM DESIGN

The algorithm that we use TF-IDF is used for term frequency (TF) in the word document. The other component is inverse document frequency (IDF) measuring the uniqueness of a word in a document. The convolutional neural networks are a subset of machine learning which contains one or more hidden layers with one output layer.

Figure 1. example of a neural network



### 3.0 Algorithm Description (Algorithm A)

To\_xy which converts a Pandas dataframe to the x,y inputs that TensorFlow needs.

Regression\_chart which takes predictions and actual outcomes and plots them on a chart.

Encode\_numeric\_range which encodes a column to a range between normalized\_low and normalized\_high.

### 3.3 Data Manipulation

#### 3.3.1 Algorithm Description (Algorithm A)

Imports specific Google drive, so that the data can be accessed.

#### 3.3.2 Algorithm Description (Algorithm B)

Reads the Amazon reviews subset JSON file to a dataframe.

#### 3.3.3 Algorithm Description (Algorithm C)

Removes unnecessary columns from subset dataframe.

#### 3.3.4 Algorithm Description (Algorithm D)

Changes image column to has image or doesn't have image 0/1

Fills empty values with 0 in overall

Removes all rows containing 0 upvotes

Adds summary test to reviewText column

Instant feedback on amazon reviews

### 3.3.5 Algorithm Description (Algorithm E)

Removes outliers from dataframe

### 3.3.6 Algorithm Description (Algorithm F)

Normalizes the vote column of the dataframe

### 3.3.7 Algorithm Description (Algorithm G)

Transforms the reviewText column into TF-IDF

## 3.4 TF-IDF Model

### 3.4.1 Algorithm Description (Algorithm A)

Prepares test train split

Creates checkpoint

defines model and loops with early stopping

Loads best model from checkpoint

Then produces regression chart

### 3.4.2 Algorithm Description (Algorithm B)

Produces a regression chart for training data

## 3.5 TF-IDF Model with Embedding

### 3.5.1 Algorithm Description (Algorithm A)

Prepares test train split

Creates checkpoint

defines model and loops with early stopping

Loads best model from checkpoint

Then produces regression chart

### 3.5.2 Algorithm Description (Algorithm B)

Produces a regression chart for training data

## 3.6 GloVe Pre-trained word embedding chart for training

### 3.6.1 Algorithm Description (Algorithm A)

Prepares uses tokenizer to fit dataframe's reviewText column

Encodes reviewText column

Loads in GloVe pre-trained model

Creates weight matrix

### 3.6.2 Algorithm Description (Algorithm B)

Prepares test train split

Creates checkpoint

defines model and loops with early stopping

Loads best model from checkpoint

Then produces regression chart

### 3.6.3 Algorithm Description (Algorithm C)

Produces a regression chart for training data

## EXPERIMENTAL EVALUATION

### 4.1 Methodology

We used the All Beauty subset of Amazon's review data spanning the range May 1996 - Oct 2018. For training and testing, we used 80% of the data for training and 20% for testing.

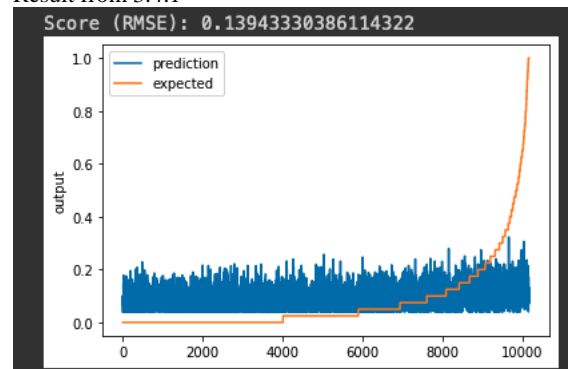
For our experimental setting we used Google Colab (a Tensorflow Python Notebook), Keras, SKLearn, Numpy, and Pandas. Using Pandas, we stripped the subset of unnecessary, combined the data in reviewText and summary, and prepared numerics for data processing. Using SKLearn, we turned the sum of reviewText and summary into TF-IDF arrays, and we separated data into a test/train split. Using Keras, we processed ut data and created our word embedding.

To compare different methods and metrics, we used a regression model and an RMSE score for comparison between the models. We also used a regression chart, so we could visualize what was happening within each model's runs.

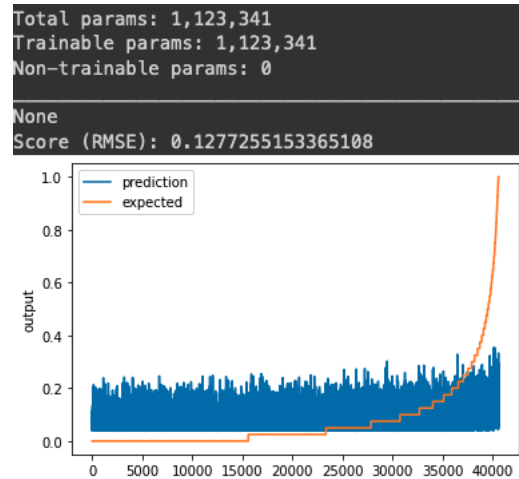
We compared TF-IDF, TF-IDF with word embedding, and GloVe pre-trained word embedding.

### 4.2 Results/Conclusion

Result from 3.4.1

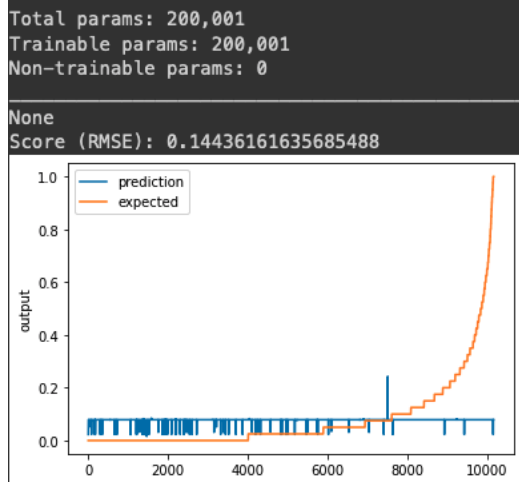


Result from 3.4.2

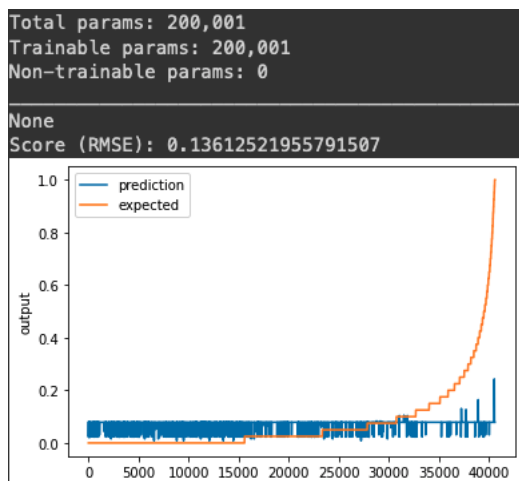


Instant feedback on amazon reviews

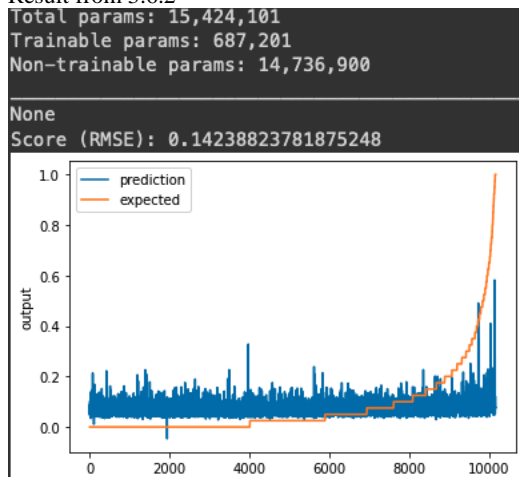
#### Result from 3.5.1



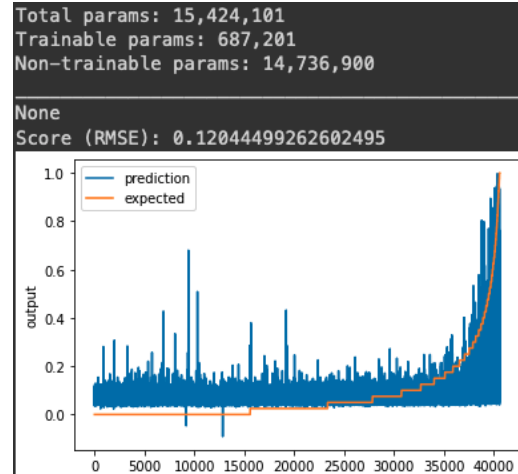
#### Result from 3.5.2



#### Result from 3.6.2



#### Result from 3.6.3



The conclusion in the results would be that the models do not train well except that GloVe, even though it had the worst RMSE score, model that doesn't perform outside of the training set.

## CONCLUSION

Despite many attempts to reconfigure the data and adjust parameters, we were unable to find a model that could predict a review that was worthy of a high helpfulness upvote on Amazon. It seems like the review text itself only plays a small part in the overall helpfulness of a review. There might be many outside influences at play here. The reviewer's friends might be upvoting the reviewer's post. The product poster might be using alternate accounts to upvote reviews that they like. It could be time based, convenience based, it could be any number of things, but it seems like we did not have the correct data for the job.

## WORK DIVISION

Brian - Code review, fully connected model with word embedding

Cody - Data preprocessing, CNN without word embedding

Kevin - Gathering results and reviewing the report, CCN with word embedding

## LEARNING EXPERIENCE

We learned how the word embedding works and what it is capable of. They are a distributed representation for text that is perhaps one of the key breakthroughs for the impressive performance of deep learning methods on challenging natural language processing problems. However, despite using such technology, the results from models cannot give the answer we were expecting. This might result in the dataset we chose had a small factor to answer our question. The helpfulness of reviews is a personal experience, and we were trying to turn subjective opinion to objective numbers.

## ACKNOWLEDGMENTS

We would like to acknowledge Professor Chen, for his suggestions on the project, as well as his guidance.

## REFERENCES

- [1] Figure 1. Provided by Professor Chen.
- [2] Pennington, J., Socher, R., & Manning, C. D. (2014, August). *GloVe: Global Vectors for Word Representation*. Glove: Global vectors for word representation. Retrieved November 2021, from <https://nlp.stanford.edu/projects/glove/>.
- [3] Ni, J. (2018). *Amazon review data*. Amazon Review Data. Retrieved November 2021, from <http://deeppyeti.ucsd.edu/jianmo/amazon/index.html>
- [4] Ruizendaal, R. (2020, June 2). *Deep Learning #4: Why You Need to Start Using Embedding Layers*. Medium. Retrieved November 2021, from <https://towardsdatascience.com/deep-learning-4-embedding-layers-f9a02d55ac12>
- [5] Cam-Stein, D. (2019, February). *Word Embedding Explained, a comparison and code tutorial*. Medium. Retrieved November 2021, from <https://medium.com/@dcameronsteinke/tf-idf-vs-word-embedding-a-comparison-and-code-tutorial-5ba341379ab0>