Classifying Amazon Product Reviews

Riya Dholakia, Caroline Liongosari, Sowmya Sridharan, Alicia Yen

Problem Statement

Can we create a model that accurately predicts a 1 to 5 star rating given a review of an Amazon product?

Example Amazon review with a star rating of 1:

Mark A

They do not have good taste, they were broken and crumbly and would ...

Reviewed in the United States on December 8, 2014

Verified Purchase

These were not even close to the quality, the price might imply. Biscotti, should be dry yes but crumbly dry no. They do not have good taste, they were broken and crumbly and would never, buy this product again.

3 people found this helpful

Helpful | Report abuse

Relevance

Analyzing customers’ sentiments on products is a core part of marketing analytics for improving products and services, thereby increasing sales and business growth

Why is our project interesting?

It goes beyond classifying positive vs negative sentiment and into more nuanced sentiments (2,3, and 4 stars), which is much trickier

Acquiring Data

- Used a dataset of Amazon fine food reviews from Stanford researchers *
- 140,000 reviews with equal # of ratings with 80% train, 10% dev, 10% test split

id	star_rating	review_body	Summary
154598	2	bought this recently and will not purchase again. for someone who likes strong cof- fee this did not do it for me	not strong enough

*Julian McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. pages 897–908

Past Works

- Taparia A. & Bagla, T (2020) first vectorized the review texts using TF-IDF and then implemented multinomial Naive Bayes, logistic regression, and linear SVC to classify 1-5 star ratings for Amazon reviews with resulting accuracies ranging from 49.2% to 54.1%
- Liu Z (2020) experimented with Count and TF-IDF vectorization on Yelp reviews and used logistic regression, multinomial Naive Bayes, and random forest getting results between 59.9% and 64.4%

Taparia, A., & Bagla, T. (2020). Sentiment Analysis: Predicting Product Reviews' Ratings using Online Customer Reviews. *Social Science Research Network*.

Baselines

	Accuracy	Avg. F1	RMSE
Average Rating	0.2	0.067	1.414
Random Rating	0.198	0.198	2.006
Naive Bayes w/ TF-IDF	0.491	0.492	1.165

Extension 1 - RNNs & CNNs

Models tried:

- LSTM (vanilla and bidirectional)
- GRU (vanilla and bidirectional)
- CNN

RNN architecture:

- Embedding layer
    - We used pre-trained word embeddings, which were fine-tuned during training
- 1 linear layer
- 1 GRU or LSTM layer
- Dropout layer

CNN architecture:

- Filter sizes e.g. [3,4,5]
- Number of filters e.g. [100,100,100] or [200,200,200]
- Dropout layer
- Max pooling

Extension 1 - RNN vs. CNN comparison

Results for the RNN and CNN architectures:

- Bidirectional LSTM/GRU outperformed their vanilla counterparts
  - Bi-LSTM is the best-performing model
- LSTM outperformed GRU in both cases
- CNN did not perform as well as the RNN models did

Model	Acc.	Avg. F1	RMSE	Training Time
LSTM	0.672	0.671	0.809	12 min
Bi-LSTM	**0.679**	**0.680**	**0.774**	**18 min**
GRU	0.660	0.658	0.787	10 min
Bi-GRU	0.667	0.666	0.783	15 min
CNN	0.627	0.627	0.928	5 min

Note: All results used minimal text preprocessing, word2vec embeddings, Adam optimizer, cross entropy loss, dropout rate of 0.3, and learning rate of 0.003

Extension 1 - Preprocessing methods

Results from comparing preprocessing methods:

- Minimal preprocessing worked the best
- Further preprocessing did not improve performance
- Additional preprocessing leads to potential information loss

Pre-processing Steps	Acc.	Avg. F1	RMSE
lowercase, remove numbers, remove punctuation, expand contractions	**0.6792**	**0.6798**	**0.7744**
lowercase, remove numbers, remove punctuation, expand contractions, remove html tags, web addresses, emojis	0.6786	0.6785	0.7795
lowercase, remove numbers, remove punctuation, expand contractions, lemmatize words	0.6761	0.6766	0.8016
lowercase, remove numbers, remove punctuation, expand contractions, remove html tags, web addresses, emojis, remove stopwords, remove POS tags	0.6536	0.6528	0.9066

Note: All results used minimal text preprocessing, word2vec embeddings, Adam optimizer, cross entropy loss, dropout rate of 0.3, and learning rate of 0.003

Extension 2 - Transformer Based Models

Experiment 1: Case vs. Uncased

Models: Bert Base Cased vs. Bert Base Uncased

Pre-processing steps:

1. Remove stopwords, punctuation, HTML tags, & web addresses
2. Expand contractions and convert emojis to text
3. Remove numbers

Tokenization: Pre-trained Bert fast tokenizer with padding and truncation

Result: Bert Base Cased performed better

Extension 2 - Transformer Based Models

Experiment 2: Filter out words with certain POS tags

Models: Bert Cased with and without POS filtering

Filtered out words with the following POS tags:
- ‘DT’, ‘PRP’, ‘CD’, ‘WDT’, ‘WP’, ‘TO’, ‘IN’, ‘CC’, ‘PRP$’, ‘WRB’

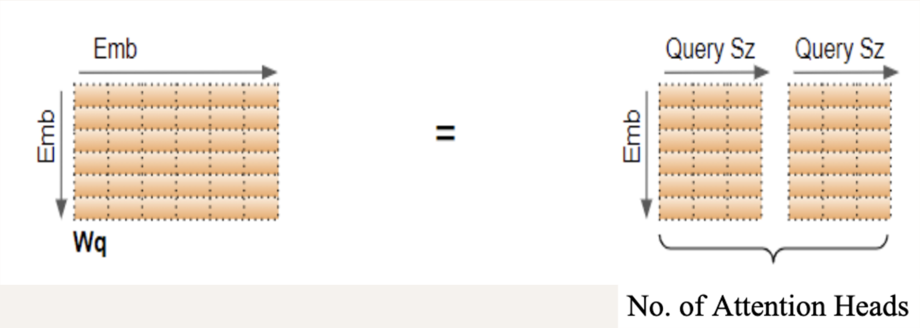
Result: Bert Cased with POS filtering performed better

Extension 2 - Transformer Based Models

Experiment 3: Varying the number of attention heads

Models: Bert Cased with 12 (default), 24, and 32 attention heads

Query Size = embedding size / number of heads



Result: Bert Cased with 12 attention heads performed best

Extension 2 - Transformer Based Models

Experiment 4: Varying type of position embeddings

Models: Bert Cased with “absolute”, “relative-key”, and “relative-key-query” position type embeddings

- Absolute: Depends on the absolute position of words in a sequence
- Relative-key and relative-key-query: Depends on the relative position between words in a sequence

Result: Bert Cased with “absolute” position type embedding performed best

Extension 2 - Transformer Based Models

Experiment 5: Varying transformer types
Models: Bert Base Cased, Distilbert Base Cased, and Roberta Base

DistilBERT

- Focused on inference speed by keeping 97% of the performance of BERT while only using half the number of parameters
- Reduced the training time to a quarter of the original amount.

RoBERTa

- Retrained BERT using dynamic masking: the masked token changes during the training epochs.
- Removed next sentence prediction objective
- Used significantly more text than BERT for training (16 GB for original BERT vs 160 GB with RoBERTa)

Result: Roberta Base performed best

Extension 2 - Transformer Based Models

Result: Roberta performed best on the development set
Test set metrics: **0.684** accuracy, **0.683** F1, **0.705** RMSE

	# of attention heads	Position Embedding Type	Accuracy	Average F1	RMSE	Training Time
BERT - cased (no POS tagging)	12	Absolute	0.668	0.663	0.745	3.5 hrs
BERT - uncased (no POS tagging)	12	Absolute	0.665	0.665	0.744	3.5 hrs
BERT - cased*	12	Absolute	0.679	0.678	0.723	3.5 hrs
BERT - cased	24	Absolute	0.667	0.792	0.667	4.5 hrs
BERT - cased	12	Relative Key	0.605	0.605	0.924	4 hrs
BERT - cased	12	Relative Key Query	0.610	0.610	0.925	4 hrs
DistilBERT - cased	12	Absolute	0.661	0.661	0.797	2 hrs
RoBERTA - cased	12	Absolute	0.680	0.678	0.739	3.5 hrs

Conclusion - Results on Test Set

	Accuracy	Avg. F1	RMSE
Average Rating	0.2	0.067	1.414
Random Rating	0.198	0.198	2.006
Naive Bayes w/ TF-IDF	0.491	0.492	1.165
RoBERTa	**0.684**	**0.693**	**0.705**
Bi-LSTM	0.676	0.677	0.783

Thank You!

Appendix

Extension 1 - Transformer Based Models

Experiment 5: Varying transformer types

Models: Bert Base Cased, Distilbert Base Cased, and Roberta Base

DistilBERT focused on inference speed by keeping 97% of the performance of BERT while only using half the number of parameters (66 million parameters vs original 110 million) and reducing the training time to a quarter of the original amount.

RoBERTa improved on BERT by changing the training methodology such that it retrained BERT using dynamic masking: the masked token changes during the training epochs. It also removed next sentence prediction objective. It also used significantly more text than BERT for training (16 GB for original BERT vs 160 GB with RoBERTa)

Result: Roberta Base performed best

Extension 2 - Word embeddings

Results from comparing pre-trained word embeddings:

- Word2Vec worked best
- All results below used the minimal preprocessing method
- All results below used a dropout rate of 0.3 and a learning rate of 0.003

Model	**Word Emb.**	**Acc.**	**Avg. F1**	**RMSE**
Bi-LSTM	Glove	0.676	0.676	0.784
Bi-LSTM	**Word2Vec**	**0.679**	**0.680**	**0.774**
Bi-LSTM	Fasttext	0.645	0.643	0.881

Error Analysis of Best Model

Category 1: Mixed messaging between product review and experience review

- The system is confused when the text combines the review for the product (“favorite taste”) with the experience (“stale cookies”)

Review	Actual	Pred.
Stale cookies cookies always favorite taste Famous Amos cookies However ordered amazon bag stale even expiration date taste super dissapointed even prevented ever buying	3	1
Melted could give gift friend loves could give gift melted plastic really stuck carmal bought waybr br dissapointed	3	2
opened package Ordered boxes Jolly Rancher candy box opened sent entired order back never order candy online	2	1

Error Analysis of Best Model

Category 2: Confusion due to possible information loss

- Text preprocessing can lead to information loss
- In row 1, “amazing” is mentioned 3 times along with “wow”
- However, something must be missing since this review has a score of 3

Review	Actual	Pred.
amazing Have putting beet grill fry amazing Took deer camp last week put fresh deer steaks wow amazing Even better sprin-kle little bit salads	3	5
Mocha Capp Breakfast Cookie individually wrapped cookies came good condition Mocha Capp favorite flavor nice change smells great toaster oven	3	4
banana pancake mix love pancakes trying different flavors kept taste buds go-ing	4	5

Error Analysis of Best Model

Category 3: Confusion between classes that share similar sentiment

- In these examples, the model is able to accurately capture sentiment
- However, it is difficult to distinguish between 4 star vs. 5 star, or 1 star vs. 2 star

Review	Actual	Pred.
husband LOVES coffee Weve tried loved flavored coffees little worried Melitta version would good costs much lessbr br pleasantly surprised coffee smooth rich blend husband loves hazelnut creamer absolutely enamored Creme Brulee Hazelnut flavor Im fan nutty-flavored coffee favorite cant wait try Melitta versions Vanilla Bean	4	5
Really bad coffee bought coffee wouldnt go way go specialized coffee shop Big mistake strong flavorful coffee expect Kona beans taste theyve sitting around months probably reality waste money better Starbucks beans	2	1

Types of position Embeddings

Attention Weights: eij

Absolute:
Input Embedding: xi = ti + si + wi , attention weight:
Absolute Position Weights: wi

$$e_{ij} = \frac{(x_i W^Q)(x_j W^K)^T}{\sqrt{d_z}}$$

Relative-Key:
Relative Position Weights: aij = w(j−i)

$$e_{ij} = \frac{(x_i W^Q)(x_j W^K + a_{ij})^T}{\sqrt{d_z}}.$$

Relative-Key-Query:
Relative Position Weights: aij = w|j−i|

$$e_{ij} = \frac{(x_i W^Q)(x_j W^K)^T a_{ij}}{\sqrt{d_z}}.$$