Predicting E-Commerce Product Recommendations

by Abigail McDonald

Background and Objective:

The project encompasses models which utilize an e-commerce dataset from a women's clothing store. Each row in the data refers to a given review on a single clothing item purchase. The data includes several demographic details on each purchase as well as a review title, review, and a recommendation (1 for yes, 0 for no). The data has been anonymized (i.e. specific retailer name has been removed). The objective of the project is to use NLP with deep neural networks to build predictive models that will indicate positive/negative recommendation.

Data Retrieval and Understanding

The women's clothing ecommerce data set is available from https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews. The review title and review columns were merged to create a single column for predictive text analysis. All other columns excluding the recommendation were dropped as this is a deep neural network problem and both text analysis and pre-trained solutions will be utilized. Additionally, rows with empty data were removed to facilitate text analysis on existing reviews. Finally, the data was split into training (80%) and testing (20%) data and normalized for use with classifier models.

Phase 2: Modeling

Data Preparation: Only the title, review and recommendation columns were of any interest and retained as only pretrained models were used. The title and review columns were combined into one column. For the CNN and LSTM models, which both utilized FastText Embeddings, the data needed to be normalized, tokenized, and padded, then the embeddings were generated and the model was fitted and evaluated. For the Universal Embedding NNLM, the data was normalized and the model was fit and evaluated. For the BERT models, the data was normalized and tokenized, the model was generated with BERT/DistilBERT embeddings and 2 dense/drop layers, as well as feature IDs, masks, and segments were created from the data. The models were then trained and evaluated.

Modeling:

Classifiers with:

- CNN with FastText Embeddings – similar in set up to LSTM, but one order of magnitude faster
- LSTM with FastText Embeddings – similar in set up to CNN, but slower

- NNLM Universal Embedding Model – longer to run than first two (relatively still fast compared to BERT models), easy set-up as embeddings are integrated into model
- BERT - The BERT model required me to lower the batch size considerably because of OOM errors
- DistilBERT – fastest to set-up and run of the BERT models, would be easiest to incorporate

Phase 3: Evaluation, Insights and Recommendations

Model performances are as follows:

Experiment No.	Model Architecture Used	Accuracy	F1 Score
1	FastText Embeddings with CNN	92.43%	92%
2	FastText Embeddings with LSTM	91.17%	91%
3	NNLM Universal Embedding	91.90%	92%
4	BERT	92.70%	92%
5	DistilBERT	92.90%	93%

The DistilBERT transformer model had the best performance and was the most cost effective to run of the transformer models. The fastest model to prepare and run with lowest system demand with moderate comparative performance was the NNLM Universal Embedding model. The recommended model is the DistilBERT model as it capitalizes on both lower resource use (compared to BERT) and highest accuracy/F1 score.

Conclusion

There were quite a few more recommendations than non-recommendations, and most of the errors happened when classifying as recommended (false positive). The DistilBert model did the best at correctly identifying the 0 class. I did initially assume the transformer models would have the best performance as they have been pretrained on large corpuses. Although, I was quite surprised that all models were within 2 accuracy points of each other. As such, in an environment with lower resources, a non-transformer DNN could be used. Ideally, however, the DistilBERT model would be the best model for balancing performance and cost.