Predicting Ratings for Local Restaurants

Aneek Das
Department of Computer Science
aneek.das@mail.utoronto.ca

Shimona Narang
Department of Chemical Engineering And Applied Chemistry
shimona.narang@mail.utoronto.ca

Abstract

With a myriad of reviews for each business, classification based on sentiment is not entirely sufficient. Information on users and products can be used in conjunction with textual data to predict the label. We compare techniques in collaborative filtering & text classification and propose a novel deep learning architecture using transformers with a stacked one-dimensional convolutional network. Our results demonstrate that our method was able to classify the review texts substantially better for skewed data. The code for all our experiments and models is available at https://github.com/aneekdas96/GoogleLocal.

1 Introduction

According to a survey [1], approximately 64% of people check reviews on Google before visiting a business, while, around 80% of people trust places that are highly rated. However, while giving feedback, reviewers frequently leave reviews without a rating or vice versa. Our study addresses this problem by helping to predict the missing ratings for given reviews. We approach this predictive task through recommender systems and text classification. The outline for this report is as follows: Section 2 introduces commonly used approaches for this task, section 3 focuses on data analysis and feature extraction, section 4 presents model architectures, while, section 5 concludes with results and discussion.

2 Related Work

Collaborative filtering is a popular technique in recommender systems where ratings are predicted through similarities between users and items. Over time, collaborative filters have developed from memory based to model based architectures. Owing to its computational inefficiency, especially in the case of sparse user-item rating matrix, latent feature subspace was introduced as an effective way for collaborative filtering. However, it only captures linear features. [2] introduces autoencoder based collaborative filtering to learn a non-linear latent feature subspace. A considerable improvement is seen in the rating predictive task.

Besides, ratings can also be predicted through multi-label text classification. [3] uses parallel 1D convolutions built on pre-trained word embeddings to predict output class probabilities. [4] covers various LSTM based architectures while [5] & [6] discuss applications of the BiLSTM with Attention architecture for sentiment classification from text. Another recent work [7] demonstrates how recommender systems can be combined with text classification techniques.

In our work, our main contribution is towards improving text classification, especially for imbalanced datasets. We employ 1D Convolutions with transformers to improve over existing models.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

3 Data Analysis and Feature Extraction

Our dataset was taken from the UCSD Recommender Systems Dataset Archive compiled by Julian McAuley [8]. The dataset contains reviews in various languages for businesses spanning across 1300 categories. We limit our study to categories in the food & drinks domain, and, a small belt in North America such that most of the reviews are in English. Overall, our sample data for study consist of 115,130 reviews.

Figure 1: **Left**: Fig 1.a shows that dataset is imbalanced **Center**: Fig 1.b shows the selected belt in North America **Right**: Fig 1.c shows distribution of review lengths

Even though our observations were constrained to the US and Canada, we found several multilingual reviews. To reduce them further, we extracted reviews that only contained ASCII. Next, the reviews were converted to lower case and all punctuations were removed. Subsequently, stop words were removed to further reduce vocabulary. We used a model pretrained on GoogleNews [9] to convert the words into 300-dimensional word vectors. For this approach, no stemmer was used as most stems were unrecognized by the pretrained model. The major advantage of using this approach was that the model ignored all words outside its vocabulary, thus filtering out most non-English words from our corpus.

4 Models

4.1 Baseline Model

As seen in [7], we combine a memory-based user-user collaborative filter, built on cosine similarities with a text classification model trained using logistic regression. The combined objective function is obtained through a weighted average function.

4.2 Stacked Convolutional Network

When operating on sequential data like text, 1D convolutions have proven to yield good results. It facilitates picking up patterns in the text which become more complex as we add more layers. The features are translationally invariant implying that the same features can be detected over different parts of the text. We used 3 1-D convolutional layers with a kernel size of 3 for each of them. 256, 128, and 64 were our chosen number of feature extractors for each layer respectively. The output features were flattened and passed through 3 fully connected layers to produces class probabilities.

4.3 Parallel Convolutional Network

Our stacked convolutional network was constrained to learning only 3-word features due to its fixed kernel size. For our parallel convolutional network, we used feature extractors of varying kernel sizes to extract structural and semantic information from word-groups of different lengths (synonymous with n-grams). Kernel sizes of 2, 3, 4, and 5 were used for our parallel convolutional layers. The outputs of these layers were then flattened, concatenated and then passed through 2 fully connected layers to get our output probabilities.

4.4 LSTM Network

Our previous models using convolutional layers do not capture long term dependencies between words. LSTMs use gates to filter and add information to the cell state while being better at handling vanishing gradients than RNNs. The inputs were then passed through an LSTM layer containing 256 cells. The outputs of these cells were then passed through 2 fully connected layers before passing through our output layer to get the class probabilities.

4.5 BiLSTM with Attention

In addition to being computationally intensive, LSTMs fail to remove the problem of vanishing gradients completely due to back-propagation through time. Attention models[10] look at the entire text at at once, making them better for learning long term dependencies. Each word is represented as a query, key, and value vector, and, similarity values are calculated between the query and the keys. Then, the softmax of the output is calculated and multiplied with the value vectors of words to get the respective attention values.

Unidirectional LSTMs only preserve information from the past. Bidirectional LSTMs combines the hidden states of the 2 LSTMs(forward and backward) to preserve information from the past and future.

The inputs were then passed through 2 Bidirectional LSTM with 20 and 256 cells respectively. The forward hidden state, forward cell state, backward hidden state, and backward cell state of the second LSTM were pairwise concatenated and fed to an attention layer with 10 attention blocks. The context vector of the attention layer was passed through 3 fully connected layers to get our predicted class probabilities.

4.6 Transformer Network

For our next model, we used a Transformer network [11] as our prediction model. Transformers consist of 2 parts. First, an encoder that uses self-attention to get the key and value vectors of words. Second, it consists of a decoder network that uses masked attention to get our query vector. Subsequently, mixed attention is used to get the word probabilities that help in determining our output class.

We used 8 attention heads for our encoder block. The output of the transformer block (word probabilities) were then passed through a Global Average Pooling layer after passing through a dropout layer. This was then fed through 2 dense layers to get our predicted class probabilities.

4.7 Transformer Block + Convolutional Network

On observing the confusion matrix for our BiLSTM + Attention Model, we found that the model was excellent in predicting 5 and 1-star reviews, while completely ignoring all other classes - implying our model's ability to learn extremely polar sentiments. We found that most 3 star reviews required learning sentiments from word combinations. To incorporate this, we passed the output word probabilities of our transformer block through 2 1-D convolutional layers with a kernel size of 3 having 128 and 64 feature extractors respectively. They were then fed into 2 fully connected layers to get our predicted class probabilities.

4.8 General Model Specifications

Each review was trimmed down to 20 words for CNN models, 40 words for LSTM & BiLSTM + Attention models and 50 words for transformer models. Shorter reviews were padded with zeros. For all models presented above, Adam was our preferred optimizer due to its ability to handle sparse gradients (very common in language processing tasks). It also incorporates adaptive learning rate per parameter that adjusts based on the magnitude of the gradients. Dropout layers were used for the models to reduce codependency between layer neurons and to reduce overfitting. In addition, for all convolutional architectures, max pooling was used to enable learning from abstract representations and to reduce the number of model parameters. Finally, categorial cross-entropy was our preferred loss function as it outputs class probabilities and allows learning from one-hot encoded class labels.

5 Results

For collaborative filtering, we were unable to get desired results due to extensive hyperparameter tuning and huge computational cost. The performance evaluation of our deep learning models was done using accuracy score and confusion metrics for each class. Figure 2 shows the confusion matrices for our top 2 performing models in terms of accuracy.

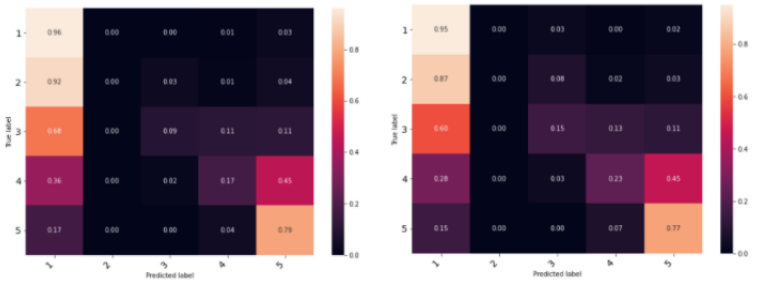

Figure 2: **Left**: Fig 2.a Confusion Matrix for BiLSTM + Attention Model **Right**: Fig 2.b Confusion Matrix for Transformer + Convolutional Network

We see that our Transformer + Convolutional Network model was able to predict classes 1 and 5 with very high accuracies, while still covering a substantial amount of 3 and 4-star reviews.
Observing the accuracies for various models, we see that the BiLSTM + Attention model had the highest accuracy on our test set as it was able to predict class 5 reviews the best. However, our preferred model is our our Transformer + Convolutional Network model due to its ability to predict 3 and 4-star reviews considerably better.

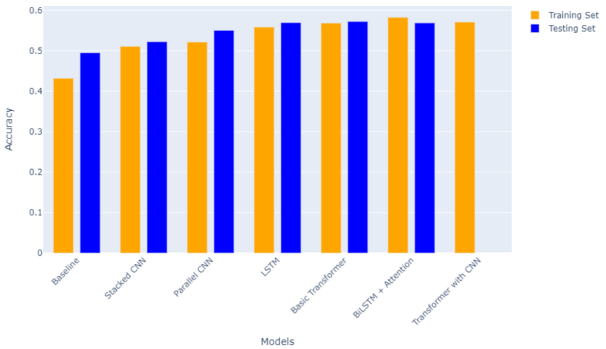

Figure 3: Accuracy for different models

6 Conclusion and Future Work

Our baseline model fails to learn the contextual features and does not generalise well with the word vectors trained from our dataset itself. Besides, both the text classifier and collaborative filters used in the baseline model capture only linear features. As an alternate approach, we stemmed the words and trained an Embedding layer to learn word vectors. However, the word vectors from our pretrained model yielded significantly better results due to availability of more data and the presence of multilingual reviews in our dataset. As expected, the vanilla LSTM model performed better than both stacked and parallel convolutional architectures due to it's ability to learn long-term dependencies. Attention based models outperformed our vanilla LSTM model due to its ability analyse the entire text and learn weighted values of words. Finally, our Transformer + Convolutional Network model was able to generalize better to average class ratings due to it's ability to learn multi-word features.

This study has potential limitations. First, due to an imbalanced dataset our models are naturally biased towards predicting class 5 ratings with a higher accuracy. Secondly, due to computational limitations, were were unable to experiment with more than 40% of the dataset while training auto-encoder based collaborative filters.

Due to their excellent ability to learn contextual relations between words (or sub-words) in a text and form efficient representations, transformer based architectures like BERT and GPT-3 are currently start-of-the-art in language modelling tasks. We hope to experiment with the above said models on our dataset.

7 Attribution

Authors contributed equally towards this project. The work was distributed as follows:

Aneek: Data Preprocessing, Stacked Convolutional Network, LSTM, Attention + BiLSTM model, Transformer Network and it's variant.

Shimona: Data Preprocessing, Exploratory Data Analysis, Collaborative Filters – Memory Based, Matrix Factorizarion and AutoEncoder, Parallel Convolutional Network

References

[1] 2018 reviewtrackers online reviews stats and survey based on reviewtrackers. `https://www.reviewtrackers.com/reports/online-reviews-survey`.

[2] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi, editors, *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, pages 111–112. ACM, 2015.

[3] Yoon Kim. Convolutional neural networks for sentence classification. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL, 2014.

[4] Jakub Nowak, Ahmet Taspinar, and Rafal Scherer. LSTM recurrent neural networks for short text and sentiment classification. In Leszek Rutkowski, Marcin Korytkowski, Rafal Scherer, Ryszard Tadeusiewicz, Lotfi A. Zadeh, and Jacek M. Zurada, editors, *Artificial Intelligence and Soft Computing - 16th International Conference, ICAISC 2017, Zakopane, Poland, June 11-15, 2017, Proceedings, Part II*, volume 10246 of *Lecture Notes in Computer Science*, pages 553–562. Springer, 2017.

[5] Qimin Zhou and Hao Wu. NLP at IEST 2018: Bilstm-attention and lstm-attention via soft voting in emotion classification. In Alexandra Balahur, Saif M. Mohammad, Véronique Hoste, and Roman Klinger, editors, *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EMNLP 2018, Brussels, Belgium, October 31, 2018*, pages 189–194. Association for Computational Linguistics, 2018.

[6] Jun Xie, Bo Chen, Xinglong Gu, Fengmei Liang, and Xinying Xu. Self-attention-based bilstm model for short text fine-grained sentiment classification. *IEEE Access*, 7:180558–180570, 2019.

[7] Bing-kun Wang, Yongfeng Huang, and Xing Li. Combining review text content and reviewer-item rating matrix to predict review rating. *Comput. Intell. Neurosci.*, 2016:5968705:1–5968705:11, 2016.

[8] Julian McAuley. Recommender systems datasets. `https://cseweb.ucsd.edu/~jmcauley/datasets.html`.

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[10] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.