**Predicting Yelp Rating Polarity:**

**Leveraging NLP for Sentiment Analysis of Local Business Reviews**

Team 9 – Indranil Pal, Yasser Parambathkandy

**George Mason University**

**AIT526 Natural Language Processing**

**Dr. Heidari**

**06/28/2023**

**Introduction**

**One of the persisting challenges in modern applications revolves around the quantification of thoughts and opinions. Despite the widespread adoption of star-based ranking systems, they often fail to capture the true essence of a reviewer's sentiment. The subjective interpretation of star ratings creates ambiguity, where a five-star rating may hold vastly different meanings for different individuals. Consequently, it becomes crucial to explore the full-text reviews to obtain a more nuanced understanding. Our project aims to bridge this gap by enabling programs to analyze and comprehend the complete review text effectively.**

**Sentiment analysis presents an invaluable opportunity to leverage the abundance of text-based data available on the internet, which comprises a significant portion of unstructured information. While humans possess the innate ability to understand textual content, programs struggle with the complexity of interpreting large volumes of text swiftly. Conversely, programs excel at processing vast amounts of data efficiently, a task beyond human capabilities. Despite significant advancements, achieving absolute accuracy in sentiment analysis remains an ongoing challenge that fuels our project.**

**This project focuses exclusively on text-based reviews extracted from Yelp.com, a prominent platform that allows users to provide feedback on businesses using a five-star rating system alongside open-ended text reviews. By scrutinizing the textual content, the limitations of simplistic star ratings can be transcended and a deeper understanding of reviewers' sentiments can be unraveled. This comprehensive analysis will enhance the evaluation process and provide a more holistic perspective on the quality of businesses.**

**Related Work**

**The various approaches and techniques in sentiment analysis, aiming to improve the accuracy and effectiveness of correlating user reviews with sentiment or ratings are as follows:**

**In their study, Yates et al. (2013) acknowledged the inherent noisiness of user reviews that are labeled with discrete numeric values. To mitigate this noise, they experimented with a different approach by categorizing the reviews simply as positive or negative, instead of assigning specific numeric values. Remarkably, this alternative labeling method achieved higher accuracy compared to traditional supervised learning algorithms that relied on the discrete numeric values. Andrew L. Maas et al. (2011) employed sophisticated classification techniques, including both unsupervised and supervised methods, to learn word vectors that captured semantic term-document information and rich sentiment content. This approach enabled the development of a classifier that could leverage continuous and multidimensional sentiment information, leading to more comprehensive sentiment analysis. K. Yessenov et al. (2009) demonstrated that even simpler techniques, such as the bag-of-words model refined with carefully selected features based on the semantics and syntactic information from the text, can yield effective results. By applying these techniques, they developed a straightforward classifier with a high success rate. In the study conducted by F. Peleja et al. (2013), it was found that Support Vector Machines (SVMs) outperformed rule-based classifiers in accurately correlating user reviews with ratings. This highlights the effectiveness of SVMs as powerful tools for sentiment analysis. Additionally, the study emphasized the importance of exploring well-documented classifiers and assessing their efficacy in the specific context. V. Narayanan et al. (2013) conducted research demonstrating that well-tailored classifiers such as Naive Bayes hold significant potential when combined with specific techniques such as effective negation handling, word n-grams, and feature selection using mutual information. By optimizing these approaches, the results achieved with Naive Bayes alone reached approximately eighty-eight percent accuracy. It is worth noting, as highlighted by Leung et al. (2008), that sentiment analysis techniques are less effective without domain-specific knowledge. The effectiveness of analyzing sentiment varies depending on the specific field or domain, such as computer product reviews versus restaurant reviews.**

**This project extends related work: This project builds upon existing research by exploring Natural Language Processing (NLP) algorithms, specifically focusing on Recurrent Neural Networks (RNNs) such as LSTM. The objective is to compare the performance of these algorithms with pre-trained models like DistilBERT and ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately). ELECTRA aims to train text encoders as discriminators rather than generators, while DistilBERT is a method for pre-training a smaller general-purpose language representation model that can be fine-tuned for various tasks, similar to larger models. The implementation of these models will be supported by PyTorch packages and other relevant tools and technologies. By utilizing these resources, the goal is to develop a robust and effective model for the project.**

**Objectives**

**The main objectives of this project encompass the following aspects:**

1. **Employing NLP models focused on sentiment analysis to estimate the polarity of Yelp ratings for local business reviews based on the corresponding textual content.**
2. **Conducting an in-depth exploration and evaluation of NLP algorithms, particularly Recurrent Neural Networks (LSTM), while comparing their performance against pre-trained models such as DistilBERT and ELECTRA.**
3. **Creating an ensemble approach that incorporates DistilBERT and ELECTRA models. This involves individually inputting the training dataset into each model and subsequently combining their outputs using a linear layer to serve as a classifier.**

**Proposed selected dataset**

**The dataset has been sourced from Kaggle, the Yelp reviews polarity dataset. This dataset is derived from the original dataset provided by the Yelp Dataset Challenge in 2015. Ratings of 1 and 2 were categorized as negative, labeled as 0, while ratings of 3 and 4 were considered positive, labeled as 1. Consequently, the dataset is structured for binary classification. It comprises a substantial amount of data, with over 560,000 training sample points and 38,000 testing sample points. Both the training and testing sample points are evenly distributed across both labels, ensuring a balanced representation. The feature column of the dataset corresponds to the review text, while the target column denotes the associated label.**

**Description of proposed system**

**Two distinct solutions are proposed for text analysis: an LSTM-based solution and a Transformer-based solution. The implementation of the LSTM model involves several steps. Firstly, the LSTM class is defined, initializing the embedding and model layers. To handle varying sequence lengths, an array is created to capture the lengths of all sequences, and padding with zeroes is performed based on the maximum sequence length. In the data preparation phase, the cleaned data is imported into pandas dataframes. The feature column of the train, validation, and test datasets is converted into lists, while the label columns are transformed into tensors. Additionally, a vocabulary dictionary is created, encompassing unique tokens from all datasets, and the maximum sequence length is determined. Using GloVe embeddings, an embedding dictionary is generated. The feature variables are then transformed into a matrix of token IDs with a column size matching the maximum sequence length. Finally, the feature lists are converted into tensors. For training, a lookup table is constructed using the vocabulary and embedding dictionaries. This table contains the embedding vectors arranged row-wise, serving as the embedding weights for the LSTM model. The ADAM optimizer and cross-entropy loss function are employed for optimization and loss calculation. Training proceeds for 10 epochs with a learning rate of 1e-3 and a batch size of 16. During each batch, the optimizer is reset to zero to prevent gradient explosion. Loss and predictions are computed for each batch, and the weights are updated through backpropagation. At the end of each epoch, the model is evaluated on the validation dataset, and it is saved if the validation accuracy improves.**

**For the Transformer-based models, specifically DistilBERT and ELECTRA, a similar code structure is followed. The first step involves preprocessing the training data, utilizing 140,000 data points divided into separate training and validation sets. Depending on the chosen model, either DistilBertTokenizerFast or ElectraTokenizerFast is used to process the data. Data loaders are created to handle the training and validation sets efficiently. Next, the selected Transformer model is initialized, ensuring proper configuration and setup. During training, the AdamW optimizer is employed along with a learning rate scheduler. The model is fed with data batches from the training loader, and the loss is calculated based on the model's outputs. The weights and biases of the model are updated accordingly. Finally, the performance of the model on the validation set is evaluated, and it is saved for future use.**

**To further enhance prediction accuracy, an ensemble approach is implemented by combining the strengths of both the DistilBERT and ELECTRA models. Individually, the training dataset is fed to each model, obtaining their respective outputs. These outputs are then passed through a linear layer acting as a classifier to merge the predictions. This ensemble technique leverages the unique perspectives and strengths of both models, resulting in improved prediction accuracy and overall robustness.**

**Proposed development platforms**

**The solution will be developed in a Windows or Linux OS with a minimum requirement of 8 CPUs and 16GB memory. The program will be developed in a conda environment utilizing software packages like nltk, pandas, pytorch, scikit-learn, transformers.**

**Baseline Solution**

**The baseline solution has been implemented using LSTM as mentioned in the proposed system. The detailed steps of the implementation is provided below:**

1. **Data Preparation: The data is first cleaned by removing stopwords, punctuation, and lemmatize the words. The zero-length strings (empty after cleaning) are removed. The data is then split into training andf validation steps. The above is repeated for the test data and saved.**
2. **Downloading GloVe embeddings: If the glove embedding file is not present in the current directory, the code attempts to download it from the Stanford NLP website.**
3. **Setting up device and random seed: The code determines the device for computation (GPU or CPU) and sets a random seed for reproducibility.**
4. **Defining hyperparameters: The hyperparameters for the LSTM model are set, including sequence length, embedding dimension, number of epochs, learning rate, batch size, hidden size, number of LSTM layers, and dropout rates.**
5. **Data preparation: The code loads the cleaned training, validation, and test data from CSV files. It processes the raw text sentences and labels, converts them into sequences of token IDs, and prepares them as PyTorch tensors.**
6. **LSTM model: The architecture of the LSTM model for sentiment analysis is defined. It consists of an embedding layer, LSTM layer(s), mean pooling layer, fully connected layer, and dropout layer.**

* **Embedding layer: This layer maps token IDs to embedding vectors. It takes the vocabulary size and embedding dimension as input parameters.**
* **LSTM layer: This layer processes the embedded sequences. It takes the input size, hidden size, number of layers, and dropout rate as input parameters.**
* **Mean pooling layer: This layer acts as a learnable weighted average over the LSTM outputs along the sequence dimension. It maps the LSTM outputs to a hidden size.**
* **Batch normalization layer: This layer applies batch normalization to the mean output of the LSTM.**
* **Output layer: This layer maps the mean output to classify positive/negative sentiment. It takes the hidden size and the number of output classes as input parameters.**

1. **Training loop: The model is trained for the specified number of epochs. In each epoch, the model parameters are updated based on the computed loss and gradients. Training loss and accuracy are calculated and displayed.**
2. **Evaluation on the test set: After training, the model parameters are loaded, and the model is evaluated on the test set. The accuracy on the test set and the confusion matrix are calculated and printed.**

**Analysis and Inferences**

**Based on the results obtained from training the LSTM model with the full training dataset, the following observations were made:**

**Dataset and Training:**

* **The full training dataset consisted of 560,000 samples.**
* **The model was trained on the entire dataset.**

**Validation Accuracy:**

* **The best validation accuracy achieved during training was 92.22%.**
* **This indicates that the model was able to generalize well to unseen data, as it performed well on the validation set.**

**Training Time:**

* **The total time required for training all epochs was 3 hours and 26 minutes.**
* **This duration provides an insight into the computational resources and time needed for training the model.**

**Overfitting:**

* **There was a decrease in validation accuracy after the first epoch.**
* **This decrease suggests that the model started to overfit the training data.**
* **Overfitting occurs when the model becomes too specialized in learning the training data and performs poorly on unseen data.**
* **To mitigate overfitting, techniques such as regularization or early stopping can be employed.**

**Test Accuracy:**

* **The test accuracy achieved using the best model (with the highest validation accuracy) was 93.63%.**
* **The test accuracy provides an estimate of the model's performance on unseen data from the test dataset.**
* **A high test accuracy suggests that the model generalized well and is capable of making accurate predictions.**

**Based on these observations, it can be inferred that the LSTM model trained on the full dataset achieved good performance with a high test accuracy. This is inline with the proposed system and the objectives of the project. However, the decrease in validation accuracy after the first epoch indicates the possibility of overfitting. Further analysis and fine-tuning of the model, such as regularization techniques or adjusting hyperparameters, may be required to improve its generalization performance and reduce overfitting.**

**Next Steps**

**Based on the results analysis, the baseline solution works as expected with high accuracy. The next steps are to develop transformer models based on DistilBERT and ELECTRA. An attempt will also be made to create an ensemble model combining the strengths of DistilBERT and ELECTRA models. The results of these models will be comnpared with the baseline LSTM model to determine the final results. These actions will provide the necessary information to predict the polarity of Yelp reviews.**

**References**

**Yelp, Inc. (2022). Retrieved from** <https://www.kaggle.com/datasets/ilhamfp31/yelp-review-dataset>

**Maas, Andrew & Daly, Raymond & Pham, Peter & Huang, Dan & Ng, Andrew & Potts, Christopher. (2011). Learning Word Vectors for Sentiment Analysis. 142-150.**

**Peleja, Filipa & Magalhães, João. (2013). Opinions in User Reviews: An Evaluation of Sentiment Analysis Techniques. 10.13140/2.1.3177.1206.**

**Leung, Cane & Chan, Stephen. (2008). Sentiment Analysis of Product Reviews.**

**Narayanan, V., Arora, I., &amp; Bhatia, A. (2013). Fast and accurate sentiment classification using an enhanced naive Bayes model. Intelligent Data Engineering and Automated Learning – IDEAL 2013, 194–201. doi:10.1007/978-3-642-41278-3\_24**

**Yang, Z. (2020). Sentiment analysis of movie reviews based on machine learning. 2020 2nd International Workshop on Artificial Intelligence and Education. doi:10.1145/3447490.3447491**

**Yates, A., Goharian, N., &amp; Yee, W. G. (2013). Semi-supervised probabilistic sentiment analysis: Merging labeled sentences with unlabeled reviews to identify sentiment. Proceedings of the American Society for Information Science and Technology, 50(1), 1–10. doi:10.1002/meet.14505001031**

**Appendix**

**A sample of the dataset from** <https://www.kaggle.com/datasets/ilhamfp31/yelp-review-dataset>

|  |  |
| --- | --- |
| Text | Label |
| Phone calls always go to voicemail and messages are not returned. Stupid way to do business. | 1 |
| Ryan Rocks! I called him this morning for some sprinkler help and some potential landscaping for my backyard. He showed up to my house within an hour and answered all of my questions! When I showed him how my irrigation box was leaking, he knew right away what to do - he went a grabbed a new part out of his truck, put it on, and it was fixed! He then set up my watering timer and we talked about some landscaping and pricing. His prices seemed great and I will definitely use him in the future! To top it all off, when I asked him what I owed him for fixing my irrigation leak and his time, he said don't worry about it! I was shocked and couldn't believe his service! Thank you so much Ryan! | 2 |
| We tried the Cheese Danish w lemon. Creamy, flaky and not overly sweet. It was fantastic. Chocolate croissant was yummy too! | 2 |
| Long line, inefficient staff. Maybe my expectations were too high but it just wasn't as good as I was hoping for the calories. We had the bread pudding and carrot cake cookie. Eh. | 1 |