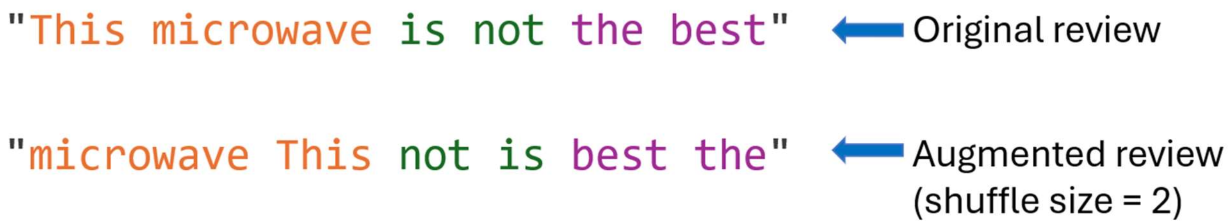
EE4940 Undergraduate Research – Final Report

Student: Nicholas Loehrke Advisor: Hynek Boril

Summary: This research focused on applying machine learning techniques to sentiment analysis of Amazon review data. I implemented a multi-stage pipeline involving preprocessing, feature extraction using FastText embeddings and recurrent neural networks (RNNs), and classification using feed-forward neural networks (FFNNs). Key topics explored were text augmentation techniques, word embeddings, RNN topologies and model evaluation metrics. The final implementation used the machine learning library PyTorch with the Python programming language to create a set of modular scripts to perform various stages of the pipeline, which were then integrated using Bash scripts. My work resulted in a functional sentiment analysis system with promising performance and many areas for future improvements.

Study Objectives and Outcomes

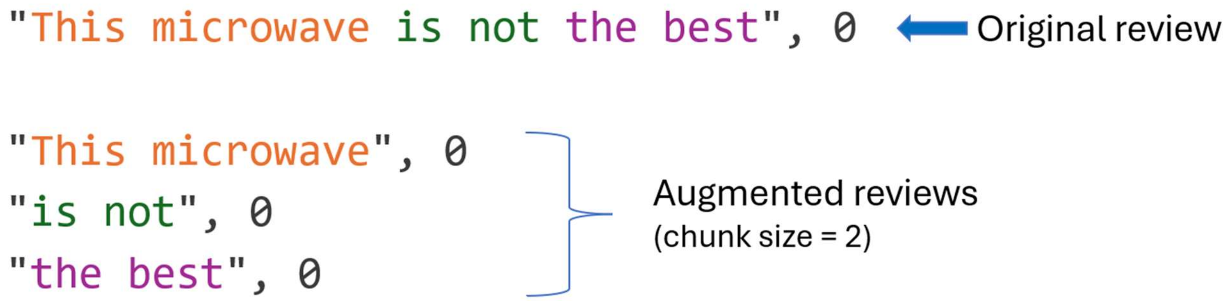
My main focus was on developing a pipeline for sentiment analysis of Amazon reviews. The first step of the pipeline implementation was preprocessing the review text. A set of Amazon reviews was collected from a previous semester's work with approximately 7000 reviews. I took these ~7000 reviews and created test and training sets and then performed two augmentations on the training set: shuffling and chunking. An example of the shuffling augmentation is shown below:



```
"This microwave is not the best"
"microwave This not is best the"
```

Shuffle augmentation.

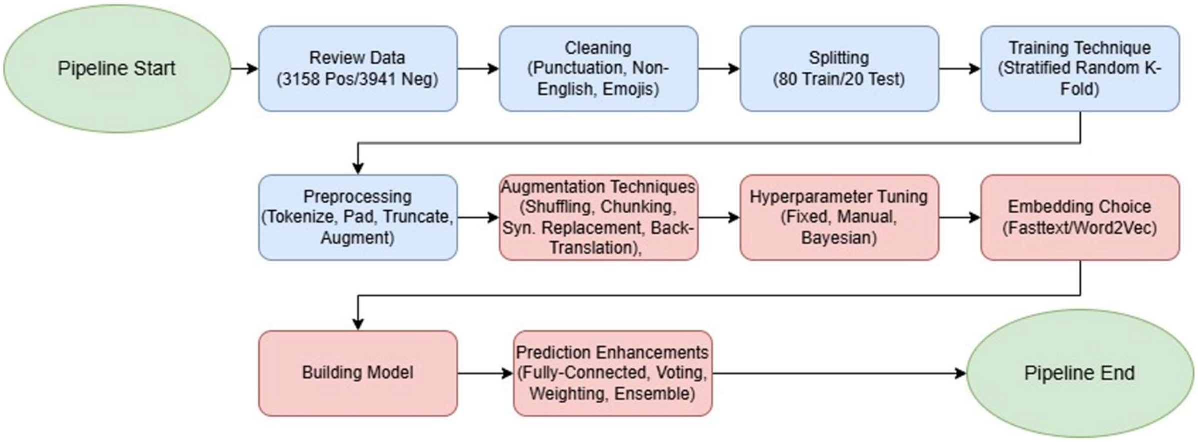
Each newly shuffled review would then be added to the set as if it were a distinct review. An example of chunking is shown below:



```
"This microwave is not the best", 0
"This microwave", 0
"is not", 0
"the best", 0
```

Chunking augmentation.

Each review chunk is also added to the set as a distinct review. These augmentations created significantly more data for the neural networks to learn from. After preprocessing and augmenting the review data, I used FastText word embeddings to convert each word from a review into a 300 dimensional vector. Word embeddings are a common technique used in natural language processing (NLP) problems as they capture semantic relationships between words. Once the review data was converted to word embeddings, an RNN was trained as a classifier and the hidden state representations of each review were extracted after training. The hidden state representations were then used to train a separate FFNN as the final classification step to determine positive and negative reviews. I found that the without augmentation, the RNN performed better than the RNN + FFNN as a review classifier, but with augmentation, the RNN + FFNN outperformed using just the RNN. The following figure shows a more detailed pipeline. Blocks in blue show parts of the pipeline that are common to myself and peers working on the project and blocks in red show where my peers and I chose unique implementations.


Pipeline Start
Review Data (3158 Pos/3941 Neg)
Cleaning (Punctuation, Non-English, Emojis)
Splitting (80 Train/20 Test)
Training Technique (Stratified Random K-Fold)
Preprocessing (Tokenize, Pad, Truncate, Augment)
Augmentation Techniques (Shuffling, Chunking, Syn. Replacement, Back-Translation),
Hyperparameter Tuning (Fixed, Manual, Bayesian)
Embedding Choice (Fasttext/Word2Vec)
Building Model
Prediction Enhancements (Fully-Connected, Voting, Weighting, Ensemble)
Pipeline End

Detailed pipeline showing common and individual implementation blocks. Blue: common, red: individual.

Hands-On Experience

A bullet list of techniques and algorithms you worked on/with, as well as software tools you used; put each technique, algorithm or software tool as a separate bullet.

- Software
  - Python
    - PyTorch
    - Pandas
    - Numpy
  - Linux
  - Bash
  - Git
- Machine learning techniques
  - RNNs
  - FFNNs
- Algorithms
  - Shuffling augmentation
  - Chunking augmentation

Presentation

Myself and two other students from UW-Platteville presented our research at the University of Texas at Dallas to the Center for Robust Speech Systems (CRSS).

Conclusion and Future Work

I achieved a modular sentiment analysis pipeline with promising performance compared to the previous semester's work using statistical methods. The use of text augmentation techniques and feature extraction using hidden state representation of reviews proved effective in increasing model performance. Future work may involve further augmentation techniques and the use of more sophisticated model architecture such as transformers.