COMP 550 Assignment 1: Text Classification

Sienna Hsu

I. Introduction

In this project, I built three linear classifiers for text: the Logistic Regression (LR) model, the Support Vector Machine (SVM) model, and the Multinomial Naive Bayes (MNB) model. Specifically, the models attempted to classify a statement about animals into "fact" or "fake." I first generated factual and fake statements about animals using ChatGPT-4 to obtain the data. I then preprocessed the data using three methods: unigram, lemmatization, and bigram. Each of the models was trained on the three datasets, producing nine results in total. The MNB model paired with lemmatization performed the best, achieving 95% accuracy. The SVM model paired with bigram performed the worst, achieving only 54% accuracy.

II. Dataset

The dataset consists of 200 facts and 200 fake statements about animals. To construct the dataset, I asked ChatGPT-4 to generate facts and fake statements about felines, canines, birds, primates, insects, and nocturnal animals. Before preprocessing, I split the dataset into 80% training data and 20% test data. The training data was turned into lowercase and removed of periods, commas, dashes, apostrophes, and stop words. Quotation marks were kept because I observed that fake statements used more quotation marks. The data was then processed in three different ways into three training datasets: unigram modeling, lemmatization, and bigram modeling. The unigram model served as the control group. Lemmatization is a unigram model with each word lemmatized; whereas the words in the bigram model did not undergo any more changes. Finally, for each training dataset, I built a vectorizer with its lexicon. The corresponding test datasets underwent the same treatments and were vectorized by the corresponding vectorizers.

III. Experiments and Results

A. Logistic Regression

For the LR models, I conducted an exhaustive grid search of the following hyperparameters using 5-fold cross-validation: inverse of regularization strength, C, (0.1, 1, 10, 100, 1000); penalty (L1, L2); and the maximum iteration during optimization, max_iter, (1, 2, 3, 4, 5, 10, 15, 50). Using the unigram model, the best LR hyperparameters are 100 for C, 1 for max_iter, and L2 for penalty; this combination produced a 93% test accuracy. The best hyperparameters for the lemmatization model are (C: 100, max_iter: 5, penalty: L1). The result also reached a 93% test accuracy. Lastly, the best hyperparameters for the bigram model are (C: 1, max_iter: 1, penalty: L2). This model performed poorly with a 68% accuracy. The low max_iter values indicate that the models overfit quickly; this was mitigated by regularization of varying strengths and penalties.

B. Support Vector Machine

For the SVM models, the hyperparameters searched are C (0.1, 1, 10, 100, 1000), kernel (linear, polynomial, rbf, sigmoid), degree for polynomial kernel (2, 3, 4), and gamma for rbf, poly, and sigmoid kernels (scale, auto). The unigram model was paired with hyperparameters (C: 1, gamma: scale, kernel: sigmoid), achieving an 89% test accuracy. Note that in the scale mode, gamma is calculated by 1/(number of features * variance of dataset). The lemmatization model reached 90% in test accuracy with C=1 and a linear kernel. The best hyperparameters for the bigram model are (C: 10, gamma: scale, kernel: rbf). The test accuracy was 54%, close to randomly guessing.

C. Multinomial Naive Bayes

Since I was working with occurrence counts (which are discrete and non-binary) of words in the

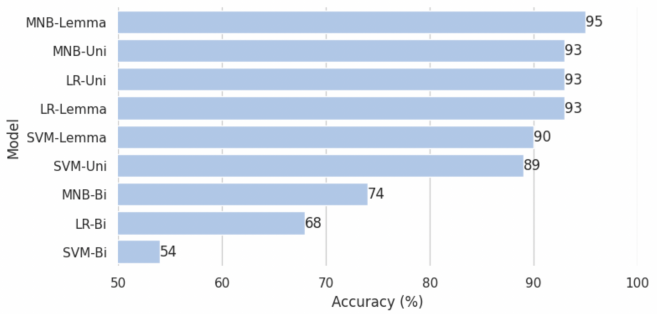

Fig. 1. Comparative results of all models

dataset, I chose to use the Multinomial Naive Bayes model over other kinds of Naive Bayes models. I tuned the hyperparameter alpha (0, 1e-12, 1e-10, 1e-08). Additionally, I experimented with giving an assigned class prior of (0.5, 0.5) over letting the model calculate the class prior based on the data points. It turned out all three MNB models performed better with the (0.5, 0.5) class prior. In scikit-learn, an alpha too close to 0 would cause numerical errors. The safeguard mechanism recommends alpha to be at least 1e-10, so I only tested the value to as low as 1e-12. All three models performed the best with alpha as 1e-12, showing that a slight smoothing helped compared to no smoothing (alpha=0). The test accuracies are 93%, 95% and 74% for unigram, lemmatization, and bigram models.

IV. Conclusion and Discission

Overall, the MNB models perform the best compared to the LR and SVM models, showing that the conditional independence assumption of words holds well. Lemmatization works the best compared to unigram and bigram modeling. The bigram model performs especially poorly for this task. The results are summarized in Figure 1.

Although the test accuracies for all three linear classifiers reached 90%, the data used for this project have some inherent limitations. Firstly, the statements about animals were generated using ChatGPT-4. Some ”factual” statements might in fact be wrong. ChatGPT-4 also generates sentences in a style according to the way it was trained. In practice, different people could phrase a statement differently. Through manual inspection, a lot of fake statements generated have similar sentence structures and patterns, which provides a hint to the classifiers. In conclusion, the text classifiers might not generalize well to real-world data.

References

I used ChatGPT-4 to generate data. All models were built with scikit-learn in reference to scikit-learn official documentation and examples.