COMP550 - Assignment 1

Enzo Benoit-Jeannin (260969262)

I. Problem setup

The focus of this investigation is the classification of animal-related facts as either real or fake, utilizing linear classifiers and a range of preprocessing techniques. Three distinct linear classifiers were explored: SVM, Logistic Regression, and Naïve Bayes, as well as the following three preprocessing techniques: lemmatization, stemming, and stop-word removal. To construct the dataset used to evaluate these models and preprocessing methods, we employed a generative AI tool.

II. Dataset Generation and procedure

To generate the dataset, 100 real and 100 fake facts about three different animals were generated using Chat GPT 3.5 [1]. Note that it was decided not to import datasets generated by other students because no prompts used for dataset generation were provided. The prompt can play a role in determining the level of difficulty in distinguishing between real and fake data, which could make interpreting the results more challenging. The first animal chosen for this dataset is the black panther, and the prompts used for its generation are as follows: "For my assignment I need to generate real and fake facts about specific animals (minimum of 100 per animals). Thus, can you provide a list of 100 fake facts about black panthers. Each fact should be roughly one to two sentences in length." and "Now generate 100 real facts about black panthers". The serpent was chosen as the second animal and the prompts were designed to generate trickier fake facts: "Now generate very tricky fake facts about serpents" and "Now in the same format, generate 100 real facts about serpents". Finally, facts about bears were generated with the intent of making questionable real facts: "Now generate 100 fake facts about bears" and "Now generate 100 facts that might seem fake about bears but that are actually true".

The experiment aimed to compare four distinct preprocessing approaches: lemmatization alone, stemming alone, both lemmatization and stop-word removal, and no preprocessing as a baseline for comparison. It was decided not to apply both lemmatization and stemming simultaneously as they are performing a similar operation. Moreover, model comparison was also conducted between SVM, Logistic Regression, and Naïve Bayes. Therefore, a total number of $4 * 3 = 12$ models were trained and compared. To better conduct this experiment, the pipeline shown on Figure 1 was designed using a set of functions.

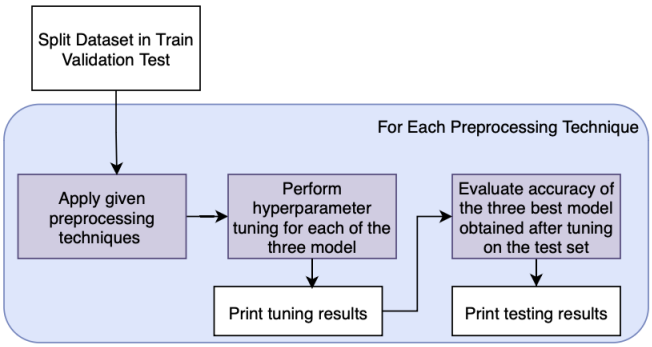

Fig. 1: Pipeline used to effectively conduct the comparisons.

To prevent data leakage and ensure rigorous evaluation, the dataset is split into training, validation, and testing sets in a 70-15-15 ratio before applying any preprocessing techniques.

III. Range of parameter settings

It is important to note that the feature matrix is created exclusively from the training set to maintain the integrity of the experiment. This process employs a Term Frequency-Inverse Document Frequency Vectorizer, which computes word frequencies within the text and normalizes them based on the total word count in each document. Although examples all have a similar lengths here, it is considered good practice to address potential variations in document length.

As all models were imported from the scikit-learn library [2], the GridSearchCV function from scikit-learn was utilized for both model training and hyperparameter tuning. The cross-validation feature within GridSearchCV was not used and instead, the concatenated training and validation sets were supplied to GridSearchCV, along with an array indicating the set in which each example belongs to (using -1 for training and 0 for validation). The GridSearchCV function trains one model per hyperparanmeter combination on each different dataset. After hyperparameter tuning, the best-performing model, determined by the highest validation accuracy score, was selected, along with the associated optimal hyperparameters. Note that all models are compared on the same metric of accuracy score.

Model	Hyperparameters List and Values
Logisitic Regression	C : $0.1, 1, 10$ Max Iter: $100, 200$
Naïve Bayes	alpha : $0.1, 1, 10$
SVM	C : $0.1, 1, 10$ Kernel: $linear$

TABLE I: Hyperparameter list and values chosen for each of the three explored classifiers.

Table I shows the selected hyperparameters each of the three classifiers. For SVM, only the linear kernel is explored, as it corresponds to the kernel type in the class material. In the context of Logistic Regression and SVM, the 'C' parameter is the regularization parameter, whereas, for the Naïve Bayes model, the 'alpha' parameter represents the Laplace smoothing parameter.

IV. Results and conclusions

All models and preprocessing techniques achieved very high accuracy on the testing set, ranging from 93% to 98%. The most effective combination was the SVM model ($C = 1$) trained on a dataset without preprocessing reaching 98%. The SVM model turned out to be the best model across all different preprocessing techniques. Both lemmatization and stemming preprocessing techniques resulted in similar accuracies, with all models achieving around 96.5% accuracy. Since stemming is less resource-intensive, it may be a more efficient option for scaling this research to a larger dataset. Conversely, the lemmatized preprocessed dataset with stop word removal showed slightly lower scores compared to datasets without stop word removal. This suggests that stop words might have relevance for distinguishing real and fake facts.

In conclusion, linear classifiers appear to be effective for distinguishing between real and fake animal facts. Furthermore, our study suggests that the choice of classifier and preprocessing techniques such as lemmatization, stemming and stop-words removal may not significantly affect the results, as all tested classifiers and processing techniques performed similarly.

V. Limitations

The study has several limitations. First, while efforts were made to make the dataset more challenging by making trickier facts, a closer examination of some examples suggests that they may not be sufficiently complex. Furthermore, the study is constrained by a relatively small dataset of only 600 examples. This research could be extended by using a larger dataset, but also by including a wider range of linear classifiers, such as Decision Trees or Random Forests, and exploring additional preprocessing techniques, including Part-of-Speech (POS) tagging.

References

[1] OpenAI, "Chatgpt 3.5 (september 25 2023 version) [large language model]," https://chat.openai.com.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.