COMP 550 - Natural Language Processing

Chaimae Sriti

September 29, 2023

1 Abstract

In this work, we empirically evaluate the performance of linear classification models in classifying between authentic and fake facts generated online. A total of 1000 data points were generated using ChatGPT-4 without verification of the factual accuracy of the purported 'facts'. The datasets exhibit a high degree of cleanliness, necessitating minimal preprocessing—comprising lowercasing, tokenization, stop word removal, and lemmatization. We optimize Logistic Regression, Support Vector Machines (SVM), and Perceptron using grid-search and cross-validation methodologies for model evaluation.

2 Experimental Setup

Only significant results will be reported in this report (run code `python a1.py` for the complete results).

2.1 Data Generation

The data was generated using various prompting strategies to provide facts about animals. Used prompts include:

- *Randomly select 10 animals and provide 20 real (or fake) facts about each, do not refer to the animal with 'they' at the begining of the fact.*
- *Provide 100 real facts about animals.*
- *Provide 100 real facts about animals.*

Data augmentation techniques such as synonym replacement were omitted. In alignment with the assignment prerequisites, an equivalent number of real and fake facts were generated, 500 for each, eliminating class imbalance issues. The datasets were saved as `fakes.txt` and `facts.txt`.

2.2 Data Preprocessing

The selected preprocessing steps include: **lowercasing**, **tokenization**, **removal of non-letter characters**, **stopwords removal** and **lemmatization**.

The evaluation will be restricted to the last three actions. To quantify the impact of each individual text preprocessing step on the overall performance, we compare scenarios where a single preprocessing action is applied against situations where no such action is taken which results in 8 combinations to evaluate.

2.3 Feature Extraction and Model Implementation

2.3.1 Data Splitting

For each dataset, we split the 1000 records into train-test stratified by the class using a 75% for training and 25% testing ratio. Other ratios were tested but only results for this one will be reported.

2.3.2 Feature extraction

Feature extraction in this work refers to text vectorization, two methods were explored in this experimentation: count vectorization and TF-IDF. Further details will be discussed in the limitations.

2.3.3 Model training and evaluation

For each combination of dataset and vectorization technique, we execute the following steps:

1. Split the data.
2. Transform the training data using the selected vectorization technique.

3. Select a model (Logistic Regression, SVM, Perceptron)
4. Use GridSearchCV to perform hyperparameter tuning for each model, identifying the best hyperparameters through cross-validation. Range of tested hyperparameters is described below:
    - Perceptron - learning rate : [0.0001, 0.001, 0.01, 0.1], regularization penalty (penalty) : ['l2', 'l1']
    - Logistic Regression - C: [0.1, 1, 2, 5, 10, 15], 'solver': ['lbfgs', 'liblinear'] with regularization penalty : l2
    - SVM - C: [0.1, 1, 10]
5. Retain the best-performing model.
6. Utilize the model to generate predictions on both the training and testing data subsets.
7. Report various performance metrics, including training accuracy, testing accuracy, and cross-validation train/test scores.

3 Conclusion

3.1 Results

In a comparative optic, among the 42 scenarios evaluated, the models yelding the best results are **logistic regression** (C=10, L2 Penalty, and lbfgs solver) and **LinearSVC** (C=1) using vectorizer **tf-idf** with a cross-validation (k=5) accuracy of 98.87% (for both models). Among the preprocessing steps, only removing stopwords has an effect as the 8 best models are all performed on datasets where stopwords were removed. Perceptron was the least performing model with a cross-validation score of 97.64%. The range of results of 42 scenarios being between 97.64% and 98.87% accuracy shows the efficiency of linear classifiers in discerning fake from authentic facts. Training and Testing accuracies are both close to 1, showing that those linear classifiers can discern almost perfectly between both classes under this experimental setting - no matter the model overall.

3.2 Limitations

We must say that this is an overly simplified problem and it is subject to an important limitation which is the nature of data, both in size and content. Data was uniformly generated from the same source, it has very limited noise and sentences obey to the same linguistic structure. Also, fake facts about animals are a relatively simple problem as those facts will generally contain words that will very rarely show in the other class. In real-world situations, data is much more messy and noisy and choosing the pre-processing actions and classification algorithms suitable might get more challenging.