Linear Classifiers for Real vs. Fake Information Detection

September 2024

1 Problem Setup

The objective of this study is to evaluate **whether linear classifiers can effectively distinguish between real and fake facts about cities**. Various linear classifiers are tested using three preprocessing methods. Following preprocessing, **unigram analysis** and **lasso regression**[3] are applied for feature extraction and selection. The best preprocessing method and hyperparameters are fine-tuned for each classifier. Finally, the classifiers are compared using the optimal settings to assess their ability to distinguish real from fake information.

2 Experiment Setup

Three linear classifiers—**Naive Bayes**, **Logistic Regression**, and **SVM**[2]—were evaluated for the task of classifying real and fake information about Hong Kong. The steps involved are outlined below.

2.1 Dataset Generation

The dataset consisted of 400 entries, split equally between real and fake facts. Both sets were generated using GPT-4 with the following prompt:

> **Give me 200 fact/fake information with length of one or two sentences describing Hong Kong.**

If there are similarities with facts or information from other cities, they are purely coincidental. No data or information has been shared with any external parties.

2.2 Models

Three models were tested: **Naive Bayes**, a probabilistic model assuming conditional independence among features; **Logistic Regression**, a linear binary classification model; and **SVM**, which seeks to maximize the margin between classes.

2.3 Preprocessing Methods

Three preprocessing methods were applied: **Lemmatization**, which reduces words to their base forms; **Stemming**, which truncates words to their root forms; and **POS Tagging**, which assigns part-of-speech labels to words. After preprocessing, the text was transformed using **TF-IDF** (unigrams), followed by **lasso regression** for feature selection to remove irrelevant features.

2.4 Hyperparameters

Two hyperparameters were fine-tuned during the experiment:

- **Split Ratio**: Tested between 0.1 and 0.3, with **0.15** performing best.
- **N-gram Range**: Tested from unigrams to 5-grams, with **unigrams** yielding the best results.

2.5 Best Performing Setup

The final experiment used the best preprocessing method, **Lemmatization** with **TF-IDF**, for the **SVM** classifier. The optimal hyperparameters were a split ratio of **0.15** and unigram n-grams[1]. Under these conditions, SVM outperformed the other models in terms of accuracy and F1 score.

3 Experiment Results and Conclusion

Three linear classifiers—Naive Bayes, Logistic Regression, and SVM—were evaluated using various preprocessing techniques and hyperparameters. The results in Table 3 show that Lemmatization was the best preprocessing method for classification, and Table 3 confirms that **Logistic Regression** was the best model for the classification task.

Model	Best Preprocessing	Validation Acc (%)	Train Acc	Test Precision (%)	F1 Score
Naive Bayes	Lemmatization	59.38	0.87	95.83	0.85
Logistic Regression	Lemmatization	59.38	0.94	92.59	0.93
SVM	Stemming	81.25	0.99	92.86	0.99

Table 1: Model Performance Results

In conclusion, linear classifiers, particularly **Logistic Regression**, proved effective in distinguishing real from fake information. The results demonstrate the importance of model selection and preprocessing techniques in improving classification performance.

Metric	Score (%)
Accuracy	80.00
Precision	92.59
Recall	71.43
F1 Score	80.65

Table 2: Best Model (Logistic Regression) Performance

4 Limitations and Future Work

The generalizability of the models in this study is limited by the **quality and simplicity of the dataset**. The **fake information** generated for this experiment is relatively short and often follows specific patterns, making it easier for classifiers to distinguish between real and fake facts based on certain **keywords or superficial features**. For instance, terms like **"international school"** frequently appeared in fake entries, which skewed the models' decisions. This **reliance on surface-level patterns** raises concerns about the models' ability to perform well on **more diverse datasets** or across different cities and topics.

Additionally, shallow models such as those used here are **vulnerable to adversarial text manipulation**. For example, altering a sentence like **"Hong Kong is known for its skyscrapers"** to **"Hong Kong is famous for its mountain ranges"** could mislead the model, as it lacks **deep semantic understanding**. These weaknesses stem from the **inherent limitations of shallow models** that rely heavily on feature patterns rather than comprehensive text analysis.

To improve robustness, future work should expand the dataset to include **more complex and nuanced fake information** across different topics. Exploring **non-linear classifiers**, such as neural networks, and employing **adversarial training methods** could further enhance performance and address the challenges posed by adversarial text.

References

[1] Peter F Brown, Vincent J Della Pietra, Peter V deSouza, Jennifer C Lai, and Robert L Mercer. *Class-based n-gram models of natural language*. Association for Computational Linguistics, 1992.

[2] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[3] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.