Building a Disinformation Classifier

Robert Gravelle

McAfee Institute

Course Number: Advanced OSINT

Abstract

In this project, we developed a binary classification model to detect disinformation in text using Logistic Regression. The dataset used was LIAR2, a comprehensive collection of political statements labeled across various levels of truthfulness. These labels were transformed into binary categories (truth vs. disinformation) for classification purposes. We preprocessed the text and applied Term Frequency-Inverse Document Frequency (TF-IDF) vectorization to convert the statements into numerical features.

The model achieved a validation accuracy of 67.5% and a test accuracy of 67.9%, with stronger performance in identifying truthful statements (precision of 0.70 and recall of 0.77) than disinformation (precision of 0.64 and recall of 0.56). To improve model accuracy, potential future work includes advanced feature extraction with word embeddings (e.g., Word2Vec, BERT), testing alternative machine learning models, and addressing class imbalance through oversampling or class weighting. This study demonstrates the value of preprocessing and vectorization techniques in building disinformation classifiers and highlights potential areas for future optimization using the LIAR2 dataset.

Building a Disinformation Classifier

# Introduction

# In today’s digital world, disinformation is a growing challenge, especially in political and social contexts. The task of detecting disinformation in text involves identifying false or misleading information and distinguishing it from truthful content. In this project, we aimed to build a binary classification model to detect disinformation in political statements using machine learning. The dataset used included labeled political statements ranging from "true" to "pants-on-fire." We transformed this multiclass data into a binary format (truth vs. disinformation) to classify statements using a Logistic Regression model.

## The primary goal was to build a machine learning model that can reliably identify disinformation in text, enabling us to understand patterns in deceptive language and misinformation. Text vectorization, data preprocessing, and appropriate model selection were key components of this task.

## Data Preprocessing

## The dataset contained raw political statements with labels indicating various levels of truthfulness. Before training the model, several preprocessing steps were applied:

## Text Cleaning: The raw text data (political statements) was cleaned by removing non-alphabetic characters, converting text to lowercase, and eliminating stopwords (common words like "the" or "and" that do not contribute significant meaning).

## Lemmatization: Words were lemmatized to reduce them to their root. This process helps standardize words and minimize variability in the data.

## Text Vectorization (TF-IDF): The cleaned text data was converted into numerical form using the Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer. This method assigns weights to words based on their frequency in a document relative to their frequency across the dataset, helping to prioritize more meaningful words over common ones.

### Vocabulary note for information. A vocabulary of the top 5000 words was selected for the model to ensure efficient feature representation while retaining relevant information.

**Model Training**

For the classification task, I chose Logistic Regression due to its simplicity and effectiveness in binary classification problems. Logistic Regression works well when the goal is to assign a probability to two possible outcomes (truth vs. disinformation as an example).

Why Logistic Regression?

Logistic Regression is a widely used algorithm for binary classification tasks. It models the probability that an instance belongs to a particular class and is interpretable, making it a good starting point for understanding disinformation classification patterns.

The multiclass labels in the dataset were mapped to binary labels:

Labels such as “true”, “mostly true”, and “half true” were mapped to the truth category (label = 1).

Labels such as “mostly false”, “false”, and “pants-on-fire” were mapped to the disinformation category (label = 0).

The model was trained on the TF-IDF vectorized training set and validated using a separate validation set to tune the model.

**Evaluation**

After training the model, it was evaluated on both validation and test sets. The key metrics used for evaluation were accuracy, precision, recall, and F1-score. These metrics provided insights into how well the model performed in identifying both disinformation and truthful statements.

* **Validation Accuracy**: 67.5%
* **Test Accuracy**: 67.9%

**Precision and Recall:**

* For disinformation (Class 0), the precision was 0.64 and recall was 0.56, indicating that the model had a harder time detecting disinformation correctly.
* For truth (Class 1), the precision was 0.70 and recall was 0.77, meaning the model was more effective at identifying truthful statements.

**F1-Score:**

* The F1-score for disinformation was 0.60, while for truth, it was 0.73. The overall weighted average F1-score was around 0.67, indicating a reasonable balance between precision and recall across both classes.

Actual results from the

Test Accuracy: 0.6794425087108014

Classification Report on Test Data:

precision recall f1-score support

0 0.64 0.56 0.60 973

1 0.70 0.77 0.73 1323

accuracy 0.68 2296

macro avg 0.67 0.66 0.67 2296

weighted avg 0.68 0.68 0.68 2296

**Challenges**

Several challenges were encountered during the project:

1. **Class Imbalance**: The dataset had more instances of truthful statements than disinformation, leading to imbalanced performance. This was evident in the lower recall for disinformation. To address this issue, we explored potential solutions like class weighting and data resampling (e.g., SMOTE).
2. **Feature Representation**: While TF-IDF vectorization performed reasonably well, it is a relatively simple method. It does not capture the contextual meaning of words, which could limit the model's ability to understand nuances in political statements.
3. **Generalization**: The model performed similarly on both the validation and test sets, but the overall accuracy of 67-68% leaves room for improvement, especially in handling more complex disinformation cases.

**Conclusion**

The disinformation classifier achieved moderate success, with an accuracy of around 68%. The model was more effective at identifying truthful statements than disinformation, highlighting potential challenges with class imbalance and feature representation.

To further improve the model, future work could explore more sophisticated techniques such as:

* **Advanced Text Representation**: Using word embeddings like **Word2Vec**, **GloVe**, or **BERT** to capture more context and semantic relationships between words.
* **Handling Class Imbalance**: Applying oversampling techniques like **SMOTE** to balance the dataset or using **class weighting** to give more importance to misclassified disinformation instances.
* **Testing Other Models**: Exploring models like **Random Forest**, **SVM**, or **XGBoost**, which may perform better on this kind of classification problem.

The project demonstrated the importance of preprocessing, vectorization, and careful model selection in building an effective disinformation detection system, and it provides a foundation for further enhancements and research.

**References**

Wang, W. Y. (2017). "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 422–426). Association for Computational Linguistics. https://doi.org/10.18653/v1/P17-2067

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830. https://jmlr.org/papers/v12/pedregosa11a.html

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781. https://arxiv.org/abs/1301.3781

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1532–1543). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162