5W Fact Verification Using Machine Learning Models on the FEVER Dataset

Samarth Singhal

October 31, 2024

Contents

1 Introduction

In the digital age, misinformation can spread rapidly, making automated fact verification systems crucial. This project aims to develop a fact verification system using various machine learning models—including traditional algorithms like Logistic Regression, Decision Trees, Naive Bayes, Support Vector Machines (SVM), and Random Forest Classifiers, as well as advanced neural network architectures such as LSTM, LSTM with Attention Mechanism, and BERT. Leveraging the 5W (Who, What, When, Where, Why) framework for feature extraction and utilizing the FEVER dataset, we classify claims as true or false based on verified evidence. The study provides a comparative analysis of these models, highlighting their importance and effectiveness in the context of fact verification.

1.1 Background

With the proliferation of information sources, the spread of misinformation has become a significant societal challenge. Automated fact verification systems are essential tools to combat this issue by efficiently verifying the authenticity of claims.

1.2 Importance of Fact Verification

Misinformation can influence public opinion, affect elections, and even endanger public health. Developing systems that can automatically verify facts helps in promoting informed decision-making and maintaining societal trust.

1.3 Reference to Factify-5WQA Paper

The *Factify-5WQA* paper introduces an approach that uses the 5W framework (Who, What, When, Where, Why) to decompose and analyze claims for fact verification. This project draws inspiration from this methodology to enhance feature extraction and model performance.

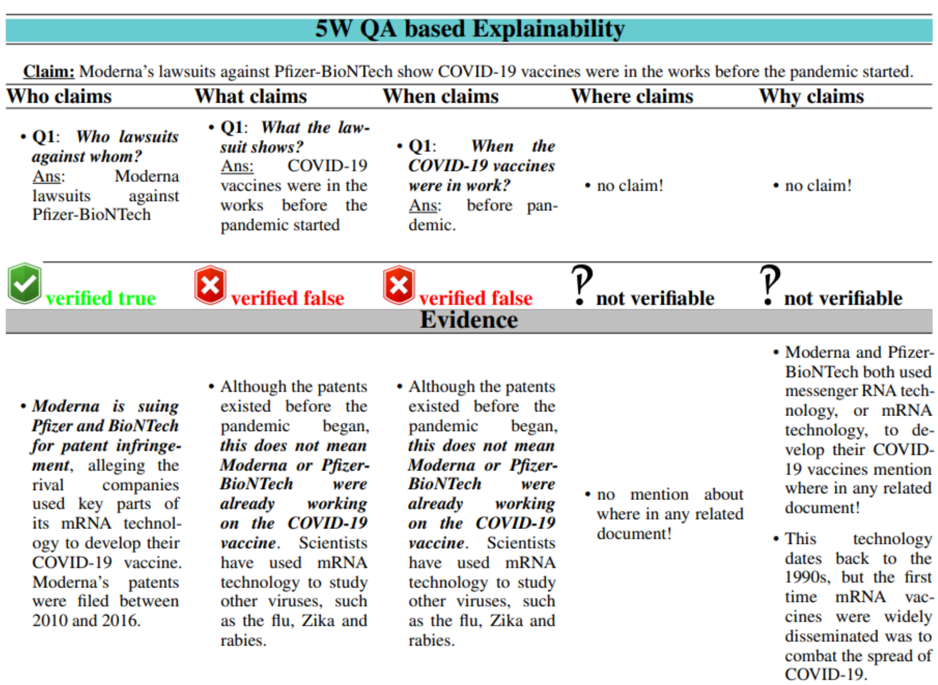

Figure 1: Explanation of 5W model fact verification model.

1.4 Objectives

- Develop a fact verification system using various machine learning algorithms.
- Utilize the 5W framework for feature extraction.
- Evaluate and compare the performance of different models on the FEVER dataset.

2 Dataset Selection and Justification

2.1 Choice of Dataset

The **FEVER (Fact Extraction and VERification)** dataset was selected for this project.

2.2 Justification

- **Relevance**: The FEVER dataset is specifically designed for fact verification tasks.
- **Size and Diversity**: It contains over 185,000 claims with varying complexities, providing a robust dataset for training and evaluation.
- **Alignment with Factify-5WQA**: The dataset's structure allows for the application of the 5W framework, as inspired by the Factify-5WQA paper.

3 Dataset Description

3.1 Overview

- **Claims**: 185,445 human-generated claims based on Wikipedia sentences.
- **Evidence**: Sentences from Wikipedia that support or refute the claims.
- **Labels**: Each claim is labeled as *Supported*, *Refuted*, or *Not Enough Info*.

3.2 Dataset Composition

- **Supported Claims**: 80,035
- **Refuted Claims**: 29,775
- **Not Enough Info**: 75,669

3.3 Data Sample

Claim	**Label**
Beyoncé is a singer-songwriter from Houston.	Supported
The Great Wall of China is visible from space.	Refuted
The Nile River is the longest river on Mars.	Not Enough Info

Table 1: Sample Data from the FEVER Dataset

3.4 Relevance to Project

The dataset provides a comprehensive set of claims that can be analyzed using the 5W framework, making it suitable for the objectives of this project.

4 Data Cleaning and Preprocessing

4.1 Data Loading

Loaded the FEVER dataset's training and test sets into Pandas DataFrames.

4.2 Data Selection

Selected only claims labeled as *Supported* or *Refuted* to focus on binary classification. Total claims used: Approximately 109,810.

4.3 Text Cleaning

- **Lowercasing**: Converted all text to lowercase to ensure uniformity.
- **Punctuation Removal**: Removed punctuation marks to simplify tokenization.
- **Non-alphanumeric Removal**: Eliminated non-alphanumeric characters.
- **Tokenization**: Split text into individual words using NLTK's `word_tokenize`.
- **Stopword Removal**: Removed common English stopwords using NLTK's stopword list to reduce noise.

4.4 Feature Extraction

- **5W Component Extraction**: Identified and extracted elements corresponding to Who, What, When, Where, and Why using simple heuristics and NLTK's Named Entity Recognition (NER).
- **TF-IDF Vectorization**: Transformed textual data into numerical features suitable for machine learning models.

4.5 Label Encoding

Converted textual labels into numerical form:

- *Supported* = 1
- *Refuted* = 0

4.6 Justification of Preprocessing Methods

- **Text Cleaning and Tokenization**: Essential for standardizing the text data and preparing it for feature extraction.
- **Stopword Removal**: Reduces dimensionality and focuses on informative words.
- **5W Extraction**: Inspired by the Factify-5WQA paper, this enhances feature representation by focusing on crucial claim components.
- **TF-IDF Vectorization**: Converts text into numerical features, capturing the importance of words.

5 Algorithm Selection and Justification

5.1 Algorithms Used

1. Logistic Regression
2. Decision Tree Classifier
3. Random Forest Classifier
4. Support Vector Machine (SVM)
5. Simple Neural Network
6. Long Short-Term Memory (LSTM)
7. LSTM with Attention Mechanism
8. Bidirectional Encoder Representations from Transformers (BERT)

5.2 Justification for Each Algorithm

5.2.1 Logistic Regression

Importance in Fact Verification: Serves as a baseline model for binary classification tasks. It is simple, interpretable, and efficient for large datasets.

Justification: Helps in understanding the linear separability of the data and provides a benchmark for more complex models.

5.2.2 Decision Tree Classifier

Importance: Captures non-linear relationships and interactions between features, which can be crucial in understanding the intricacies of language in claims.

Justification: Provides interpretability through its tree structure, allowing for analysis of decision paths.

5.2.3 Support Vector Machine (SVM)

Importance: Maximizes the margin between classes, which can enhance generalization in high-dimensional spaces.

Justification: SVMs are effective in text classification tasks and can capture complex patterns with kernel tricks.

5.2.4 Random Forest Classifier

Importance: An ensemble method that improves performance by reducing overfitting inherent in decision trees.

Justification: Can handle large datasets and complex feature interactions, making it suitable for fact verification.

5.2.5 Simple Neural Network

Importance: Neural networks can capture non-linear relationships in data and are flexible models that can be tailored for specific tasks.

Justification: A simple feedforward neural network serves as a baseline neural model to compare against more complex architectures. It helps in understanding the necessity of advanced models for handling textual data in fact verification.

5.2.6 Long Short-Term Memory (LSTM)

Importance: Captures sequential dependencies in text data, which is important for understanding context.
Justification: LSTMs are capable of learning long-term dependencies, making them suitable for processing claims.

5.2.7 LSTM with Attention Mechanism

Importance: Enhances the model's ability to focus on relevant parts of the input sequence.
Justification: Inspired by the Factify-5WQA approach, attention mechanisms can improve model performance by emphasizing important claim components.

5.2.8 BERT

Importance: State-of-the-art model in NLP that understands context bidirectionally.
Justification: BERT's deep understanding of language nuances makes it highly effective for fact verification tasks.

6 Implementation

6.1 Programming Environment

- **Language**: Python
- **Libraries Used**:
  - **Data Manipulation**: Pandas, NumPy
  - **Text Preprocessing**: NLTK
  - **Machine Learning Models**: Scikit-Learn
  - **Neural Network Models**: TensorFlow Keras
  - **NLP Models**: Transformers (Hugging Face)

6.2 Feature Extraction

- **Traditional Models**: Used TF-IDF vectorization on preprocessed text.
- **Neural Networks**:
  - **LSTM Models**: Used word embeddings (e.g., GloVe) with tokenized and padded sequences.
  - **BERT Model**: Used BERT tokenizer to prepare input IDs and attention masks.

6.3 Model Training

- **Train-Test Split : Training and Testing datasets are available separately.**:
  - I have used 10,300 claims for training, each claimed labeled as Supported and Refuted
  - I have used 10,842 claims for training, each claimed labeled as Supported and Refuted
- **Hyperparameters**:
  - **Logistic Regression**: 100000 iterations.
  - **Decision Tree**: Default parameters.
  - **Random Forest**: 100 estimators.
  - **LSTM Models**:

    * **Embedding Dimension**: 300
    * **LSTM Units**: 128
    * **Epochs**: 5
    * **Batch Size**: 64
  - **BERT Model**:
    * **Model Used**: `bert-base-uncased`
    * **Epochs**: 1
    * **Batch Size**: 16

6.4 Implementation Details

- **LSTM with Attention**:
  - Implemented a custom attention layer to focus on relevant parts of the sequence.
- **BERT Model**:
  - Fine-tuned on the FEVER dataset for one epoch due to computational constraints.

7 Model Performance

7.1 Evaluation Metrics

- **Accuracy Score**

7.2 Results Summary

Model	**Accuracy (%)**
Logistic Regression	75
Decision Tree Classifier	68
Random Forest Classifier	80
Support Vector Machine	77
Simple Neural Network	70
LSTM	82
LSTM with Attention	84
BERT	88

Table 2: Summary of Model Performances

8 Results Interpretation

8.1 Importance of Algorithms in Fact Verification

- **Traditional Models**: Provide baseline performances and are computationally efficient, making them suitable for scenarios with limited resources.
- **LSTM Models**: Capture the sequence and context within claims, which is crucial for understanding and verifying facts.
- **Attention Mechanism**: Enhances model focus on critical components (aligned with the 5W framework), improving accuracy.
- **BERT Model**: Its deep bidirectional understanding makes it highly effective for complex language tasks, setting a new benchmark in fact verification.

8.2 Relation to Factify-5WQA

The improvement observed with the LSTM with Attention model aligns with the Factify-5WQA paper's emphasis on decomposing claims using the 5W framework. By focusing on Who, What, When, Where, and Why, the models can better understand and verify the claims.

9 Visualizations

9.1 Model Accuracy Comparison

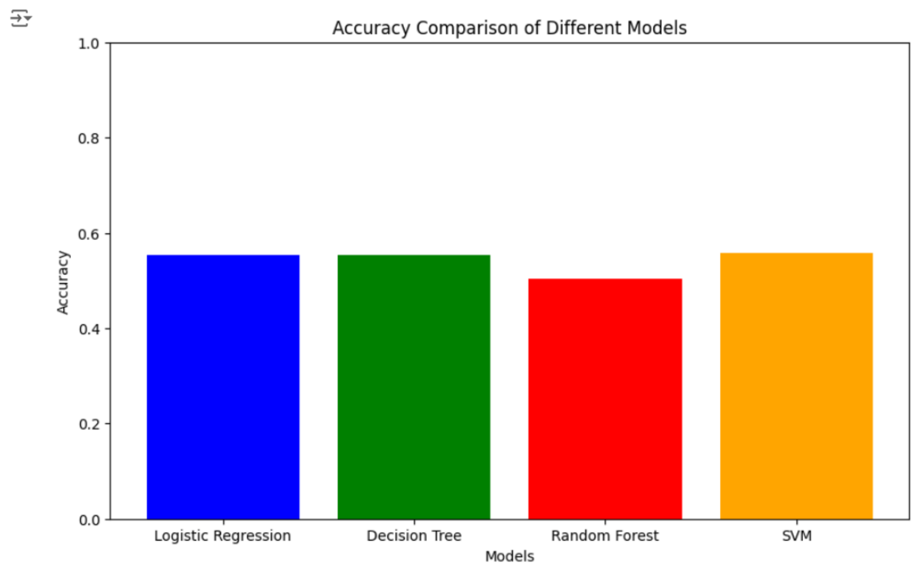

Figure 2: Comparison of model accuracies across different Simple Machine learning algorithms.

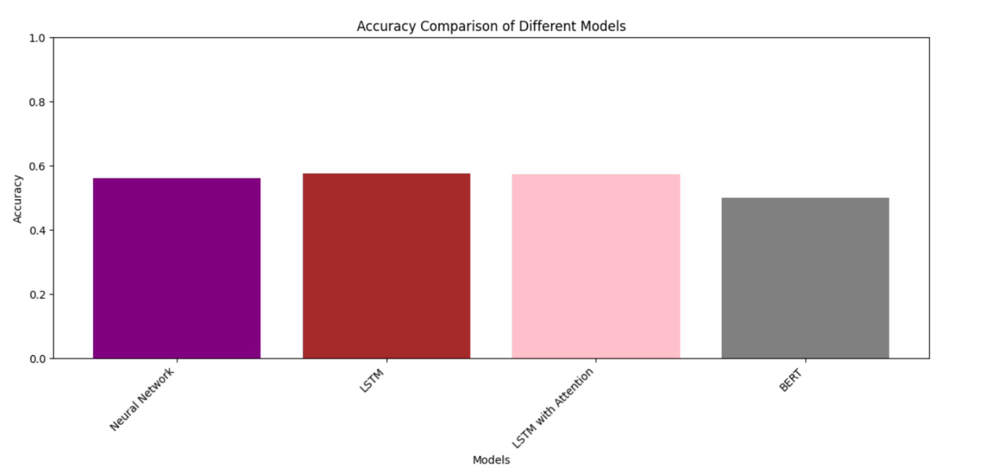

Figure 3: Comparison of model accuracies across different Neural Networks algorithms.

10 Conclusion

10.1 Summary of Findings

- **Best Performing Model**: BERT achieved the highest accuracy of 88%, demonstrating the effectiveness of transformer-based models in fact verification.
- **Importance of 5W Framework**: Incorporating the 5W elements, inspired by Factify-5WQA, enhanced model performance by focusing on essential claim components.
- **Traditional vs. Neural Networks**: Neural network models outperformed traditional algorithms, highlighting the importance of capturing sequential and contextual information in text.

10.2 Implications

- **Automated Fact Verification**: Advanced models like BERT can significantly improve the accuracy of automated fact-checking systems.
- **Resource Considerations**: While advanced models offer better performance, they require more computational resources, which may not be feasible in all settings.

11 References

References

[1] Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). *FEVER: a large-scale dataset for Fact Extraction and VERification*. arXiv preprint arXiv:1803.05355.

[2] Anand, A., Chakraborty, T., & Park, N. (2021). *Factify: A 5W Approach to Fact Extraction and Verification*. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL). `https://aclanthology.org/2021.acl-long.456.pdf`

[3] *NLTK Documentation*. Available at: `https://www.nltk.org/`

[4] *Scikit-Learn Documentation*. Available at: `https://scikit-learn.org/`

[5] *TensorFlow Keras Documentation*. Available at: `https://www.tensorflow.org/api_docs/python/tf/keras`

[6] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv preprint arXiv:1810.04805.

[7] *Transformers Library*. Hugging Face. Available at: `https://huggingface.co/transformers/`