News Title-Body Consistency Checker using LLM

by
Priya Kumari(2247008)
Ch. Mohan Santhi (2246025)
Mani Kanta (2206281)
under guidance of
Dr. Bhaskar Mondal
(Assistant Professor)

Department of Computer Science and Engineering
National Institute of Technology Patna

March 28, 2025

Table of Contents

1. Problem Statement
2. Objective
3. Tech Stack
4. Dataset Description
5. Summarization
6. Semantic Similarity
7. Classification Model
8. Workflow Diagram
9. Conclusion
10. References

Introduction

What is News Title-Body Consistency?

In today's digital era, misleading headlines are widespread, influencing public perception and spreading misinformation. This project leverages Large Language Models (LLMs) to analyze the alignment between news titles and their respective bodies, ensuring factual consistency.

Why is this important?

- Misinformation can manipulate opinions and decisions.
- Automated verification enhances trust in news sources.
- LLMs provide scalable and efficient fact-checking.

Table of Contents

1. Problem Statement
2. Objective
3. Tech Stack
4. Dataset Description
5. Summarization
6. Semantic Similarity
7. Classification Model
8. Workflow Diagram
9. Conclusion
10. References

Problem Statement

The project aims to detect whether a news title accurately represents its article. Fake or misleading titles can spread misinformation, so our AI model helps verify their authenticity.

Table of Contents

1. Problem Statement
2. Objective
3. Tech Stack
4. Dataset Description
5. Summarization
6. Semantic Similarity
7. Classification Model
8. Workflow Diagram
9. Conclusion
10. References

Project Objective

- Develop a framework to analyze the relationship between news titles and body content.
- Implement a fine-tuned LLM for generating concise news summaries.
- Compute semantic similarity between the summary and title using NLP techniques.
- Train a classifier to detect misleading news based on extracted linguistic and statistical features.

Table of Contents

1. Problem Statement
2. Objective
3. Tech Stack
4. Dataset Description
5. Summarization
6. Semantic Similarity
7. Classification Model
8. Workflow Diagram
9. Conclusion
10. References

Tech Stack

- **Programming Language:** Python (Efficient for ML and NLP applications)
- **Key Libraries:**
  - **Transformers** - For handling LLM-based models
  - **Sentence-BERT** - For computing semantic similarity
  - **Scikit-Learn** - For training classification models
  - **Pandas & NumPy** - For efficient data handling
- **Models Used:**
  - BART - Pretrained Transformer model for text summarization
  - Sentence-BERT - Embedding-based similarity measurement model

Table of Contents

1. Problem Statement
2. Objective
3. Tech Stack
4. Dataset Description
5. Summarization
6. Semantic Similarity
7. Classification Model
8. Workflow Diagram
9. Conclusion
10. References

Dataset Description

- Dataset consists of several news articles from reputable and unreliable sources.
- **Dataset Link:** Click here to access the dataset.
- Contains labeled data with three essential attributes:
  - **Title** - Headline of the news article
  - **Body** - Full content of the news article
  - **Label** - Classification (Real/Fake)
- Data preprocessing includes:
  - Removal of HTML tags and special characters.
  - Tokenization and text normalization.
  - Handling of stopwords and redundant spaces.

Table of Contents

1. Problem Statement
2. Objective
3. Tech Stack
4. Dataset Description
5. Summarization
6. Semantic Similarity
7. Classification Model
8. Workflow Diagram
9. Conclusion
10. References

Summarization using LLM

- Fine-tuned BART model on a dataset of news articles to generate concise summaries.
- Summaries help extract the core idea of the article, reducing the amount of text for comparison.
- Example Output:

  Original Body: *A detailed news article about an economic crisis...*

  Generated Summary: *Brief explanation highlighting the economic crisis.*

Table of Contents

1. Problem Statement
2. Objective
3. Tech Stack
4. Dataset Description
5. Summarization
6. Semantic Similarity
7. Classification Model
8. Workflow Diagram
9. Conclusion
10. References

Semantic Similarity Analysis

- Used Sentence-BERT to generate numerical representations (embeddings) of the title and summary.
- Computed cosine similarity score between title embedding and summary embedding.
- High similarity indicates the title accurately represents the content.
- Example Scores:
    - **Consistent Article:** 0.89
    - **Misleading Article:** 0.45

Table of Contents

1. Problem Statement
2. Objective
3. Tech Stack
4. Dataset Description
5. Summarization
6. Semantic Similarity
7. Classification Model
8. Workflow Diagram
9. Conclusion
10. References

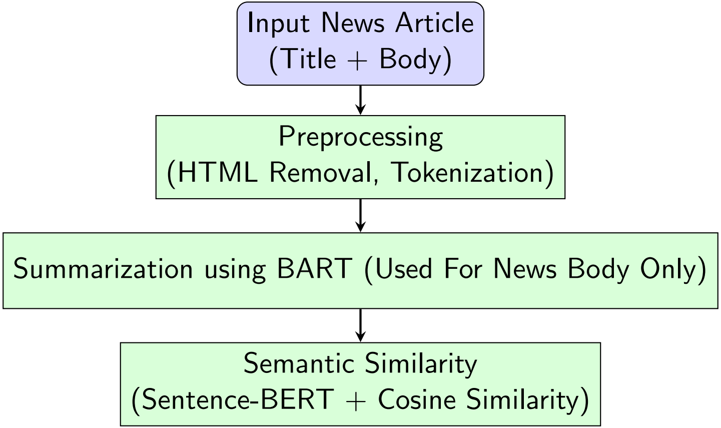
Title-Body Classification

- Trained a Random Forest classifier using:
  - Semantic similarity score
  - Sentiment polarity of title and body
  - Word count distribution
- Model outputs a prediction of whether the news article is Real or Fake.

Table of Contents

1. Problem Statement
2. Objective
3. Tech Stack
4. Dataset Description
5. Summarization
6. Semantic Similarity
7. Classification Model
8. Workflow Diagram
9. Conclusion
10. References

Workflow Diagram - Part 1



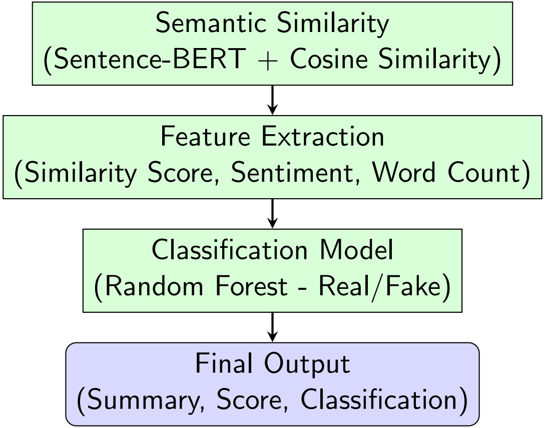
Input News Article
(Title + Body)

Preprocessing
(HTML Removal, Tokenization)

Summarization using BART (Used For News Body Only)

Semantic Similarity
(Sentence-BERT + Cosine Similarity)

Workflow Diagram - Part 2



Semantic Similarity
(Sentence-BERT + Cosine Similarity)

Feature Extraction
(Similarity Score, Sentiment, Word Count)

Classification Model
(Random Forest - Real/Fake)

Final Output
(Summary, Score, Classification)

Table of Contents

1. Problem Statement
2. Objective
3. Tech Stack
4. Dataset Description
5. Summarization
6. Semantic Similarity
7. Classification Model
8. Workflow Diagram
9. Conclusion
10. References

Conclusion

- Implemented an AI-based approach to analyze the consistency between news titles and their body content.
- Utilized NLP techniques to assess semantic similarity and detect misleading headlines.
- The system automatically checks if news titles match their content, reducing the need for manual checking.

Table of Contents

1. Problem Statement
2. Objective
3. Tech Stack
4. Dataset Description
5. Summarization
6. Semantic Similarity
7. Classification Model
8. Workflow Diagram
9. Conclusion
10. References

Reference Links

- Reference 1: Natural Language Processing with Python – Steven Bird, Ewan Klein, Edward Loper
- Reference 2: Deep Learning for Natural Language Processing – Palash Goyal, Sumit Pandey, Karan Jain
- Reference 3: Speech and Language Processing – Daniel Jurafsky, James H. Martin

Thank You!