Misinformation Detection Framework

This repository contains a multi-component framework for detecting misinformation in social media, which consists of the following modules:

[1] Temporal Characteristic Model
[2] User Profiling
[3] Causal Effect Estimation
[4] Misinformation Propensity Prediction
[5] Semantic Similarity Analysis
[6] Argument Mining and Logical Fallacy Detection (AMLF)
[7] Sentiment Analysis

The main objective of this framework is to detect misinformation and identify factors contributing to its propagation. Each module is designed to perform a specific task that contributes to the overall performance of the misinformation detection system.

Getting Started

These instructions will help you set up the environment and install the required dependencies for running the framework.

Prerequisites

- Python 3.7+
- PyTorch 1.9+
- transformers 4.10+
- spaCy 3.1+
- textblob 0.15.3+
- scikit-learn 0.24+
- pandas 1.3+

✦ Copy code
✦ pip install -r requirements.txt
✦ Usage

The framework can be used to detect misinformation by passing text samples through the different modules. Each module is designed to extract specific features and insights that contribute to the overall misinformation detection process.

To use the framework, import the corresponding Python scripts for each module and run the provided functions with the required parameters. The output of each module can be combined and processed further to make final predictions.

For more information on how to use each module, please refer to the documentation within the corresponding Python scripts.

Temporal Characteristic Model (temporal_model.py)

This module focuses on capturing the temporal characteristics of information dissemination in social media, such as time intervals, interaction ratios, and BFS ratios.

- Usage

  To use the Temporal Characteristic Model, import the corresponding Python script and run the provided functions with the required parameters.

For more information on how to use this module, please refer to the documentation within the corresponding Python script.

User Profiling (user_init.py, ppr.py, user_profiling.py)

The User Profiling module aims to capture the propagation patterns of misinformation by analyzing user profiles and their interactions in social networks.

- Usage

  To use the User Profiling module, import the corresponding Python script and run the provided functions with the required parameters.

Causal Effect Estimation (user_causal_model.py)

This module focuses on estimating the causal effects of user attributes on misinformation propagation, helping to design more effective intervention strategies.

- Usage

  To use the Causal Effect Estimation module, import the corresponding Python script and run the provided functions with the required parameters.

The Misinformation Propensity Prediction module aims to predict whether a given user is likely to propagate misinformation, enabling targeted interventions.

- Usage

  To use the Misinformation Propensity Prediction module, import the corresponding Python script and run the provided functions with the required parameters.

For more information on how to use this module, please refer to the documentation within the corresponding Python script.

Semantic Similarity Analysis (ssa.py)

This module employs a novel Semantic Similarity Analysis (SSA) approach using a pre-trained RoBERTa model, multi-layer attention mechanism, and contrastive learning to enhance early misinformation detection.

- Usage

  To use the Semantic Similarity Analysis module, import the corresponding Python script and run the provided functions with the required parameters.

For more information on how to use this module, please refer to the documentation within the corresponding Python script.

Argument Mining and Logical Fallacy Detection (AMLF) (amlf.py)

The AMLF module is designed to identify argumentative structures and logical fallacies in textual data, helping to detect inconsistencies and flaws in the reasoning of misinformation.

- Usage

  To use the Argument Mining and Logical Fallacy Detection module, import the corresponding Python script and run the provided functions with the required parameters.

For more information on how to use this module, please refer to the documentation within the corresponding Python script.

Sentiment Analysis (sentiment_analysis.py)

The Sentiment Analysis module aims to classify the sentiment expressed in text samples as positive, negative, or neutral, helping to detect potential misinformation based on sentiment patterns.

- Usage

  To use the Sentiment Analysis module, import the corresponding Python script and run the provided functions with the required parameters.

For more information on how to use this module, please refer to the documentation within the corresponding Python script.

Acknowledgments

- RoBERTa models
- Hugging Face Transformers library
- spaCy NLP library
- TextBlob library