Fake News Detection: Advances and Architectures

Yassin Taoumi

February 19, 2025

Outline

Unsupervised WhatsApp Fake News Detection using Semantic Search

Technologies Used

- **Selenium WebDriver:** For scraping WhatsApp messages from the web interface.
- **Newspaper3k Python Library:** For extracting news articles from credible sources.
- **Google Translate API:** For translating multilingual messages into English.
- **Gensim Python Package:** For text summarization and keyword extraction.
- **Sentence Embedding Models:** BERT, RoBERTa, and DistilBERT for semantic similarity comparison.

Techniques Employed

- **Web Scraping:** Extracting WhatsApp messages and news articles from the web.
- **Data Cleaning and Preprocessing:** Removing emojis, links, and translating messages.
- **Text Summarization and Keyword Extraction:** Extracting key information from messages and articles.
- **Opinion vs. Claim Classification:** Using a probabilistic LSTM model to classify messages.
- **Semantic Similarity Search:** Using BERT, RoBERTa, and DistilBERT to compare the meaning of messages and articles.
- **Cosine Similarity:** Measuring the similarity between sentence embeddings.

Model Performance

Model Name	Accuracy
BERT Models	
bert-base-nli-stsb-mean-tokens	66.67 %
bert-large-nli-stsb-mean-tokens	**78.09 %**
RoBERTa Models	
roberta-base-nli-stsb-mean-tokens	68.41 %
roberta-large-nli-stsb-mean-tokens	71.57 %
DistilBERT Model	
distilbert-base-nli-stsb-mean-tokens	73.16 %

Table: Accuracy of different models on fake news detection.

Fake News Detection WhatsApp Bot

Technologies Used

- A Twilio account
- A Twilio WhatsApp sandbox
- Python 3
- Flask
- ngork
- Tensorflow
- LIAR Dataset

Steps

1. Preprocessing:
   - Reading and cleaning the data.
   - Tokenizing and stemming.
   - Exploratory data analysis.
2. Feature Selection:
   - Bag-of-words and n-grams.
   - TF-IDF weighting.
   - Word2Vec and POS tagging.
3. Classification:
   - Naive Bayes, Logistic Regression, Linear SVM, Stochastic Gradient Descent, and Random Forest classifiers.
   - Comparing F1 scores and confusion matrices.
   - Parameter tuning using GridSearchCV.
4. Prediction:
   - Using the selected model (Logistic Regression) for classification.
5. Integrating Twilio WhatsApp API:
   - Building a Flask API server.
   - Generating an endpoint using ngrok.

Results

```
n-grams & tfidf confusion matrix and F1 scores

#Naive bayes
 [841 3647]
 [427 5325]
 f1-Score: 0.723262051071

#Logistic regression
 [1617 2871]
 [1097 4655]
 f1-Score: 0.70113000531

#svm
 [2016 2472]
 [1524 4228]
 f1-Score: 0.67909201429

#sgdclassifier
 [  10 4478]
 [  13 5739]
 f1-Score: 0.718731637053

#random forest
 [1979 2509]
 [1630 4122]
 f1-Score: 0.665720333284
```

Figure: n-grams & tfidf confusion matrix and F1 scores.

MultiProSE: A Multi-label Arabic Dataset for Propaganda, Sentiment, and Emotion Detection

Introduction

- Propaganda is a form of persuasion used to influence people's opinions.
- Resources for Arabic propaganda detection are limited.
- MultiProSE is the first Arabic dataset for multi-label propaganda, sentiment, and emotion detection.
- It extends the existing ArPro dataset with sentiment and emotion annotations.
- The dataset contains 8,000 annotated news articles.

Dataset Details

- Collected from various Arabic news domains.
- Annotated for propaganda, sentiment, and emotion.
- Propaganda labels: True/False.
- Sentiment labels: Positive/Negative/Neutral.
- Emotion labels: Happiness/Sadness/Anger/Fear/None.

Annotation Process

- Manual annotation by three native Arabic speakers with doctoral degrees.
- Annotation guidelines were provided and reviewed by experts.
- Quality control mechanisms were used to ensure accuracy.
- Inter-annotator agreement was measured using Light's Kappa and Fleiss' Kappa.

Experiments and Results

- Baselines were established using AraBERT, XLM-ROBERTa, and GPT-40-mini.
- AraBERT outperformed other models in propaganda detection.
- GPT-40-Mini achieved the highest score in sentiment analysis.
- GPT-40-Mini also performed well in emotion detection.

Model Performance

	AraBERT			XLM-RoBERTa		
Task	Micro-F1	Macro-F1	Acc %	Micro-F1	Macro-F1	Acc %
Propaganda Detection	**0.769**	**0.756**	77	0.683	0.597	68
Sentiment Analysis	0.736	0.722	73	0.698	0.682	69
Emotion Detection	0.675	0.635	67	0.648	0.608	64

	GPT-40-Mini		
Task	Micro-F1	Macro-F1	Acc %
Propaganda Detection	**0.769**	0.733	76
Sentiment Analysis	**0.842**	**0.825**	**84**
Emotion Detection	**0.750**	**0.707**	**75**

Table: MultiProSE results on test set.

LLM-enhanced Multiple Instance Learning for Joint Rumor and Stance Detection

Introduction

- Misinformation on social media is a growing concern.
- Rumor detection and stance detection can complement each other.
- Existing methods require post-level stance annotations, which are costly.
- This paper proposes a weakly supervised approach using only claim-level labels.

Model Overview

- Uses an undirected microblog propagation model.
- Transforms the multi-class problem into multiple MIL-based binary classification problems.
- Employs a discriminative attention layer to aggregate outputs.
- Leverages LLMs to capture complex interactions between post pairs.

Model Architecture

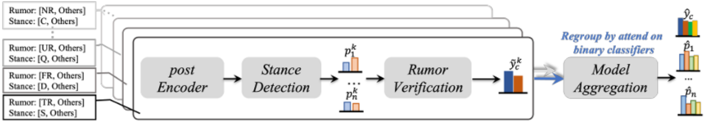

Figure: A framework of LLM-enhanced weakly supervised propagation model.

Model Architecture

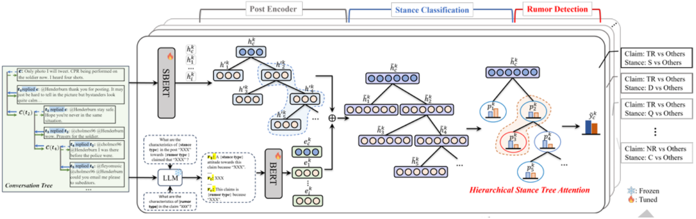

Figure: More Detailed Model Representation.

Model Architecture

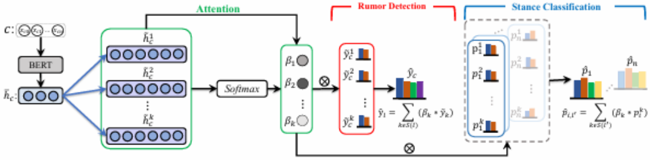

Figure: Aggregation Model Architecture

Post Encoder

- Enhances post representation using bottom-up/top-down tree transformers.
- Leverages LLMs (e.g., ChatGPT) to generate stance explanations.
- Combines post-level and explanation-level representations.

MIL-based Binary Classification

- Each binary classifier focuses on a specific veracity-stance pair.
- Stance classification predicts the binary stance probability of a post.
- Rumor classification aggregates stances using hierarchical attention.

Binary Models Aggregation

- Employs claim explanation-guided attention to combine predictions from binary classifiers.
- Achieves final multi-class prediction for both stance and rumor.

Experiments and Results

- Evaluated on three rumor datasets and two stance datasets.
- Demonstrates strong performance in joint rumor and stance detection.
- Shows promising results compared to state-of-the-art methods.

Bridging the Gap: LLM-Generated Social Context

- **Challenge:** LLM-enhanced MIL relies on real social media data with propagation structures, which may not always be available or sufficient.
- **Solution:** Generate synthetic social context using LLMs, as explored in "Let Silence Speak."
- **Benefits:**
    - Create diverse and comprehensive datasets for training rumor and stance detection models.
    - Control the distribution of stances and rumor types to address data imbalance issues.
    - Simulate various scenarios and user behaviors to improve model robustness.

LLM-Generated Comments as Synthetic Social Context

- "Let Silence Speak" demonstrates the potential of LLMs to generate realistic comments.
- These comments can be used to construct synthetic conversation threads and propagation trees.
- This provides a valuable resource for training LLM-enhanced MIL models without relying solely on real social media data.

Large Language Model Agent for Fake News Detection

Technologies Used

- **Pre-trained LLMs:** The core of FactAgent, used for natural language understanding and generation.
- **LangChain Framework:** Facilitates interaction with the LLM and external tools.
- **gpt-3.5-turbo:** The specific LLM employed in the analysis engine.
- **SerpAPI:** Enables web searching and retrieval of conflicting information.
- **External Knowledge Database:** Stores past experiences and verified URLs for credibility assessment.

System Architecture

1. **News Claim Input:** FactAgent receives a news claim, including title, domain URL, and publication date (if available).
2. **Structured Expert Workflow:** The LLM follows a predefined workflow to analyze the news claim from multiple perspectives.
3. **Evidence Collection:** Each tool provides observations based on its analysis.
4. **Final Verification:** The LLM compares the collected evidence against an expert checklist to determine the veracity of the news claim.
5. **Output:** FactAgent provides the final prediction (real or fake) with a detailed explanation of the reasoning process.

Specialized Tools

- **Internal Knowledge Tools:**
  - Phrase_tool: Identifies sensationalism, emotional language, or exaggeration.
  - Language_tool: Detects grammar errors, misused quotes, or excessive capitalization.
  - Commonsense_tool: Assesses the reasonableness of the claim against common sense.
  - Standing_tool: Checks for political bias and promotion of specific viewpoints.
- **External Knowledge Tools:**
  - Search_tool: Uses SerpAPI to find conflicting reports from other sources.
  - URL_tool: Evaluates the credibility of the domain URL using internal and external knowledge.

LLM-Enhanced multimodal detection of fake news

Awesome Fake news