A Preprocessing and Embedding Framework for Social Media Fake-News Classification

Rohit Viswakarma Pidishetti
nfrac.india@zohomail.com

***Abstract*—This study outlines a complete preprocessing and text-embedding pipeline designed for fake-news classification on social media datasets. The workflow includes text normalization, stopword removal, lemmatization, and vectorization using a transformer-based sentence embedding model. The processed dataset and learned embeddings are exported for downstream machine-learning experimentation.**

***Index Terms*—Fake-news detection, social media, text preprocessing, sentence embeddings, transformer models**

I. Introduction

The rapid spread of misinformation on social media has increased the need for automated classification systems capable of distinguishing fake content from legitimate news. Text preprocessing and high-quality vector representations are fundamental to the success of such models. This work focuses on constructing a robust preprocessing pipeline followed by embedding generation using a state-of-the-art `SentenceTransformer` model.

II. Dataset

The raw dataset (_74429.csv) contains two primary fields:

- **news** — Text content of social-media posts
- **label** — Class label indicating real or fake information

The dataset is loaded using Pandas, and the entire pipeline is applied to the `news` column.

III. Methodology

A. Preprocessing

A multi-stage normalization workflow was implemented.

1) Stopword Compilation: A large, manually curated stopword list was constructed. It includes:

- Standard English stopwords
- Contraction variations (e.g., `doesnt`, `wasnt`, `cant`)
- Context-specific noise tokens frequently present in social-media text

2) Text Cleaning Function: Each sentence s_i is cleaned using the following operation:

$$T_i = \text{Lemmatize}\Big(\text{RemoveStopwords}\Big(\text{Lowercase}\big(\text{RemoveSpecialChars}(s_i)\big)\Big)\Big) \tag{1}$$

where T_i is the cleaned text ready for embedding.

Each processed sentence is appended to a new Pandas DataFrame:

```
data_frame = pd.DataFrame({
    "news": data['news'].apply(cleanse),
    "class": data['label']
})
```

3) Saving the Cleaned Dataset: A cleaned version of the dataset is exported as _74429_V01.csv. This file represents the finalized textual input for embedding generation.

B. Sentence Embedding Generation

After preprocessing, the cleaned dataset is passed to a transformer-based embedding model.

1) Model Selection: The model used is `SentenceTransformer: all-mpnet-base-v2`, known for delivering high-quality semantic embeddings suitable for classification tasks.

2) Embedding Computation: Each cleaned text sample T_i is encoded into a dense vector E_i:

$$E_i = f_\theta(T_i) \tag{2}$$

where f_θ is the transformer-based embedding model.

3) Exporting Embeddings: Two `.pkl` files are generated:

- `embeddings.pkl` — Stores all vector embeddings
- `embedding_classes.pkl` — Stores the corresponding class labels

C. Optional Classification

For downstream classification using a linear classifier:

$$\hat{y}_i = \sigma(WE_i + b) \tag{3}$$

where $\hat{y}_i$ is the predicted probability of fake/real, W is the weight matrix, b is bias, and σ is the sigmoid function.

IV. Results

The outcome of this workflow includes:

- A noise-free, lemmatized dataset ready for ML tasks
- High-dimensional semantic embeddings suitable for downstream classification
- Clean separation between text processing and model training
- Reusable serialized embedding objects

This pipeline reduces preprocessing overhead and ensures reproducibility for experiments on fake-news detection.