Propaganda Detection using Transformers and CNN-LSTM

Team Members:

Aly Farrag Elmandouh

Amira Ali Mohamed

Aya Adel Saied

Propaganda detection (Binary & Multi-label)

The notebooks demonstrate a binary & multi-label text classification model built using two techniques Transformers and combination of CNN and LSTM layers.

The process includes data preprocessing, model building, training, evaluation, and problem-solving steps encountered during the project.

Contents:

1- Dataset:

The dataset we used was ArPro which is constructed to be the largest dataset for the task, in scale of or larger than datasets in multiple languages, dataset is based on two collections of such articles: AraFacts, and a large-scale in-house collection contains true and false Arabic claims verified by fact-checking websites, and each claim is associated with online sources propagating or negating the claim, the data sets contain multiple version but we worked on the binary and multilabel versions.

2- Data Preprocessing:

Data preprocessing was shared across all the models with differences between binary and multilabel datasets.

- **Binary Preprocessing**: The dataset is preprocessed by cleaning the paragraphs using various techniques such as normalizing Arabic text, removing diacritics, punctuations, stop words, non-Arabic letters and converting labels to binary.

- **Multilabel Preprocessing**: applied same technique as binary preprocessing in addition to implementing MultiLabelBinarizer which convert the labels into a binary format, where each label is represented as a binary vector this step will make the labels ready as an input for the training

3- Model Building:

- **Transformers model**: The transformers architecture used was Arabertv2 which is the latest version and it is designed for natural language understanding tasks. It employs self-attention mechanisms to capture contextual relationships between words in a sentence
- **CNN_LSTM model**: The model architecture combines CNN and LSTM layers. The CNN part consists of an Embedding layer, two Convolutional layers with ReLU activation and Max Pooling, followed by two LSTM layers. The final layer is a Dense layer with sigmoid activation function to predict the probability of each label. To prevent overfitting, a Dropout layer is added after the LSTM layers with a dropout rate of 0.4.

4- Model Training:

- **Transformers**: after cleaning process datasets was tokenized using Bert base tokenizer, then data is then converted into a format compatible with the Hugging Face Dataset class, the model is trained using the Trainer class from the transformer's library, which handles the training loop, evaluation, and optimization. The model's performance is evaluated using metrics like precision, recall, and F1 scores (both micro and macro).
  Multilabel custom trainer: class inherits from the Trainer class and overrides the compute_loss method. This allows the loss computation to consider multiple labels, used to handle the unique requirements of predicting multiple labels for each instance. This involves several key adjustments like Data Formatting, Loss Calculation, Evaluation Metrics, Training Configuration and Handling Undefined Metrics

- **CNN-LSTM**: The model is trained using binary cross-entropy loss, which is suitable for classification. The Adam optimizer is used with default learning rate. We used experimental parameters without fine-tuning for the binary model it trained for 5 epochs with a batch size of 32 while in multilabel model the training process is performed for 3 epochs larger batch size of 64 [to reduce training time].

5- Model Evaluation:

The model's performance is measured by loss, accuracy, precision, recall, and F1-score with micro average for the multilabel data providing a comprehensive assessment of its effectiveness on the test set.

Transformers

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Binary	0.74	0.81	0.76	0.78
Multilabel	0.36	0.71	0.41	0.52

CNN_LSTM

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Binary	0.65	0.72	0.71	0.72
Multilabel	0.17	0.53	0.33	0.41

6- Challenges and Solutions:

- **Problem:** Loading large JSONL files efficiently.
  Solution: Implemented efficient reading by loading line-by-line to avoid memory issues.

- **Problem:** Handling Arabic text involves specific challenges such as diacritics, various forms of letters, and punctuation.
  Solution: Developed comprehensive text cleaning functions to normalize, remove diacritics, punctuations, and stop words, and to handle non-Arabic letters efficiently.

- **Problem**: long training time on transformers
  Solution: Training the model on GPU using pytorch library with cuda enabled.

- **Problem**: Transformer model is not returning a loss value in multilabel training which is necessary for training
  Solution: Modify the model to return logits suitable for multi-label classification. Adjust the compute_loss method to use BCEWithLogitsLoss.

- **Problem**: Classification metrics can't handle a mix of multilabel-indicator and binary targets
  Solution: Adapted compute_metrics to handle multilabel classification by using prediction threshold to convert the logits to binary labels.
- **Problem:** Text sequences had varying lengths, which could lead to issues during model training.
  Solution: Used padding to ensure uniform sequence length across all samples, setting the maximum length based on the longest sequence in the training data.

- **Problem:** Low test accuracy.
  Solution: Further tuning of hyperparameters and exploration of additional regularization techniques may be needed.

- **Problem:** Finding the optimal combination of hyperparameters was challenging.
  Solution: Conducted grid search and cross-validation to identify the best set of hyperparameters

7- References:

- Kaur, Harpreet, and Ranjit Singh. "Text Preprocessing: A Review." International Journal of Computer Applications 975 (2016). This paper reviews various text preprocessing techniques used in NLP.
- Alharbi, Alaa, and Mark Lee. "Kawarith: an Arabic Twitter Corpus for Crisis Events." Proceedings of the Sixth Arabic Natural Language Processing Workshop. 2021. This paper discusses datasets relevant to Arabic NLP tasks.
- Hochreiter, Sepp, and Jürgen Schmidhuber. "Long Short-Term Memory." Neural Computation 9.8 (1997): 1735-1780. This foundational paper discusses LSTMs and their applications.
- Hasanain, Maram, Fatema Ahmed, and Firoj Alam. "Can GPT-4 Identify Propaganda? Annotation and Detection of Propaganda Spans in News Articles." arXiv preprint arXiv:2402.17478 (2024).
- Keras Documentation: https://github.com/fchollet/keras
- NLTK Documentation
- MultiLabelBinarizer — scikit-learn 1.5.1 documentation
- Pytorch-cuda Documentation