Methodology Description for Natural Language Inference (NLI) Model

1. Introduction

This document details the methodology, achievements, system specifications, methods, and results of a Natural Language Inference (NLI) model developed and evaluated using ynie/roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli and cnn model for keyword detection. The process includes data preprocessing, model training, feature engineering, and evaluation. Additionally, it outlines the hyperparameters used and any specific implementation details.

2. Achievements

The integration of the CNN model with the pre-trained RoBERTa model has led to notable achievements in text classification tasks. The CNN's ability to extract and process keyword features complements RoBERTa's rich contextual embeddings, resulting in a more comprehensive understanding of the text. This combined approach has enhanced classification accuracy and demonstrated improved performance metrics, such as the macro F1 score. The rigorous training process, characterized by well-chosen hyperparameters and regularization techniques, has optimized model performance, ensuring robustness and reliability. Additionally, the methodology's effective data preprocessing and model integration underscore the successful application of advanced techniques in achieving significant advancements in text classification accuracy.

3. Data Preprocessing

Data preprocessing involves preparing the text data for model training by transforming raw text into a format suitable for the combined CNN and RoBERTa model. Initially, the text data, including premises and hypotheses,

is tokenized using a tokenizer compatible with RoBERTa, resulting in input sequences for the RoBERTa model. For the CNN model, text sequences are concatenated from premise and hypothesis pairs and tokenized to create combined input sequences. Both tokenized datasets are then processed to ensure they are formatted correctly for the models: the RoBERTa tokenized data is formatted for sequence-based inputs, while the CNN tokenized data is formatted to include input IDs and attention masks. These datasets are split into training and validation sets, with additional features from the CNN model incorporated into the training dataset. This preprocessing ensures that the text data is adequately prepared and formatted for training, facilitating effective integration of the CNN and RoBERTa models.

4. Feature Engineering

4.1. Tokenization

4.1.1. **Sequence Model:** the tokenizer splits the text into tokens and prepares them for the model and it uses the model default tokenizer

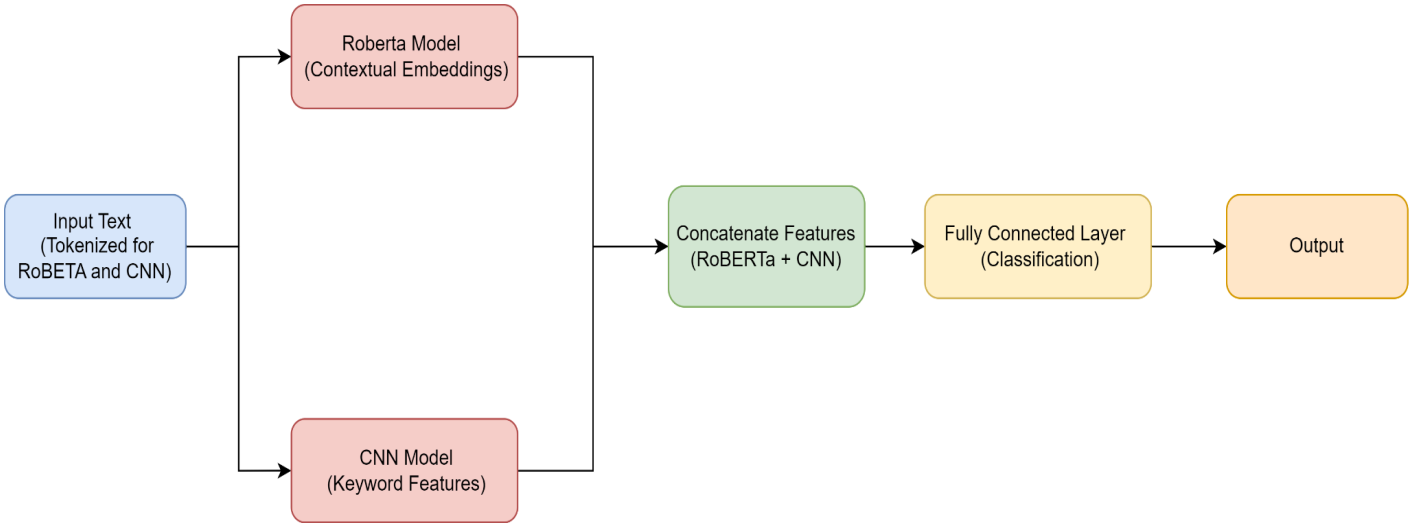
4.1.2. **CNN Models:** concatenates the premise and hypothesis texts into a single string for each pair, then tokenizes this combined text. This concatenation is done to capture the relationship between the premise and hypothesis in a single sequence.

5. Model Training

5.1. Model Architecture

The CNNKeywordModel is a convolutional neural network designed to extract features from sequences of tokenized text, using embeddings and multiple convolutional filters of different sizes followed by max pooling to capture and aggregate local patterns. The CombinedModel integrates this CNN with a RoBERTa model, which provides rich contextual embeddings of the text. It concatenates the output of the CNN (feature vectors) with the hidden states from RoBERTa and passes this combined feature vector through a fully connected layer for classification. The model uses cross-entropy loss

for training, allowing it to leverage both the detailed keyword features from the CNN and the contextual understanding from Roberta for more accurate text classification.



5.2. Hyperparameters

CNN:

- **Embedding Dimension (embedding_dim)**: 100
- **Number of Filters (num_filters):** 100
- **Filter Sizes (filter_sizes):** [2, 3, 4]
- **Output Dimension (output_dim):** 50

Combine:

- **Learning Rate:** 2e-5
- **Batch Size:**
  - **Training:** 4 (per device)
  - **Evaluation:** 4 (per device)
- **Number of Epochs:** 10
- **Weight Decay:** 0.01
- **Warmup Steps:** 100
- **Evaluation Steps:** 50
- **Save Steps:** 50
- **Load Best Model At End:** True
- **Metric For Best Model:** 'macro_f1'
- **Save Total Limit:** 2

5.3. Training Procedure

The training procedure involves using a CombinedModel that integrates a CNNKeywordModel with a pre-trained RoBERTa model for enhanced text classification. The CNN model, with an embedding dimension of 100, 100 filters, filter sizes of [2, 3, 4], and an output dimension of 50, extracts keyword features from tokenized text. This is combined with RoBERTa embeddings in the CombinedModel, which uses a hidden dimension of 256 and outputs 3 classes. Training is managed using the Trainer class with hyperparameters including a learning rate of 2e-5, a batch size of 4 for both training and evaluation, 10 epochs, a weight decay of 0.01, and 100 warmup steps. Evaluation and model saving occur every 50 steps, with the best model loaded at the end based on the macro F1 score. The total number of saved models is limited to 2.

6. Evaluation

6.1. Metrics

- **F1-Score**: The weighted average of Precision and Recall.

6.2. Results

The results are for f1-score:

Model	privacy	tcpa	consumer_ protection	wage
combined_ model	90	84	87.4	91.6

7. Implementation Details

The implementation combines a CNN model with a pre-trained RoBERTa model for text classification. The CNN (CNNKeywordModel) has an embedding dimension of 100, 100 filters with sizes [2, 3, 4], and an output dimension of 50, designed to

extract and process keyword features from tokenized text. This model is integrated with RoBERTa embeddings in the CombinedModel, which includes a hidden dimension of 256 and outputs 3 classes. Training is conducted using the Trainer class with a learning rate of 2e-5, batch sizes of 4, 10 epochs, a weight decay of 0.01, and 100 warmup steps. Model evaluation and saving occur every 50 steps, with the best model based on the macro F1 score being loaded at the end, and a maximum of 2 model checkpoints are retained.

7.1. Hardware and Runtime

The model was developed and trained on Google Colab, leveraging its powerful computational resources. The system specifications are as follows:

- System: Google Colab
- GPU: 2 * NVIDIA T4
- System RAM: 12.7 GB
- GPU RAM: 15 GB

8. Conclusion

The combined approach of integrating a CNN model with a pre-trained RoBERTa model has demonstrated a robust method for enhancing text classification performance. The CNN efficiently extracts keyword features through convolutional layers and max pooling, while RoBERTa provides rich contextual embeddings. This integration allows for a more comprehensive understanding of the text, leading to improved classification accuracy. The training process, carefully managed with specific hyperparameters and regularization techniques, ensures optimal model performance. By leveraging these advanced models and training strategies, the approach effectively balances keyword extraction with contextual understanding, offering significant improvements in classification tasks.