Ikhlaq Ahmad

Dr. Tatiana Erekhinskaya

CS 6320

May 6, 2024

Machine-Generated Text Detection: Another Step Towards Academic Integrity

**Introduction:**

Machine-generated text is not a new dimension in artificial intelligence. Historically, many such applications like chatbots Eliza have been a subject of experimentation with one goal in mind: Human-like language generation. Recent advances in the machine learning and artificial intelligence domains have made it possible to achieve human-level accuracy in automated language generation. With higher level of accuracy and nearly undifferentiability texts from language models, it is now a serious concern for originality and integrity of work, especially in academia.

**Goal:**

The goal of this project is to develop a model capable of distinguishing whether a given text is authored by a human or generated by an artificial intelligence (AI) system. Text authorship detection has various practical applications, including content filtering, plagiarism detection, and identifying fake news. In this project, we employ a Convolutional Neural Network (CNN) architecture for text classification. CNNs have proven to be effective in natural language processing tasks, especially in tasks like text classification, sentiment analysis, and language translation.

**Dataset:**

For training and evaluation, we use a dataset from Kaggle.com named AI\_Human.csv containing texts authored by humans and texts generated by AI systems. The dataset is pre-processed to ensure uniformity in text length and format. Each text sample is labeled as either 0 for Human or 1 for AI. The dataset contains around 480000 rows of predetermined essays. **A screenshot of a computer screen

Description automatically generated**

A screenshot of a document

Description automatically generated

**Model Architecture:**

The Convolutional Neural Network (CNN) architecture chosen for this text authorship detection task is designed to effectively capture and learn hierarchical features from textual data. The architecture comprises several layers arranged in a sequential manner:

* **Embedding Layer:** The input text data was first passed through an embedding layer, which converted the text into dense vector representations called word embeddings. These embeddings captured semantic relationships between words and enable the model to learn contextual information.
* **Convolutional Layers:** Convolutional layers consists of multiple filters or kernels that slide across the input embeddings to extract features. Each filter performs a convolution operation, capturing different patterns or features present in the text. The use of multiple filters with varying sizes allows the model to learn both local and global features.
* **Activation Function:** After each convolution operation, an activation function (such as ReLU - Rectified Linear Unit) is applied elementwise to introduce non-linearity into the model, enabling it to learn complex relationships within the data.
* **Max Pooling Layers:** Following the convolutional layers, max pooling layers were employed to down sample the feature maps obtained from the convolutional operations. Max pooling retains the most significant information from each feature map while reducing the dimensionality of the data, which helps in preventing overfitting and improving computational efficiency.
* **Flatten Layer:** The output feature maps from the max pooling layers are flattened into a one-dimensional vector, which serves as the input to the subsequent fully connected layers.
* **Dense Layers:** Fully connected dense layers are added to the model to perform classification based on the learned features. These layers enable the model to learn complex decision boundaries and mappings between the input features and output labels.
* **Output Layer:** The final layer of the network is a single neuron with a sigmoid activation function, which produces a probability score indicating the likelihood of the input text being authored by a human or generated by an AI system. The output value close to 1 signifies high confidence in Human authorship, while a value close to 0 indicates high confidence in AI authorship.

The overall architecture of the CNN model is designed to efficiently process textual data, automatically learn relevant features, and make accurate predictions regarding the authorship of the input text. By leveraging the hierarchical nature of convolutional operations, the model can effectively capture patterns and relationships within the text, enabling it to discriminate between human and AI-generated content.

**Training**

Training the Convolutional Neural Network (CNN) model for text authorship detection involves several key steps aimed at optimizing the model's parameters and improving its performance on the task. The training procedure can be outlined as follows:

* **Data Preprocessing:** Before training the model, the dataset was preprocessed to ensure uniformity and compatibility with the model architecture. This preprocessing included tokenization, padding, and vectorization of the text data.
* **Data Splitting:** The dataset was divided into three subsets: training, validation, and test sets. The training set was used to train the model, the validation set was used to tune hyperparameters and monitor performance during training, and the test set was used to evaluate the final performance of the trained model.
* **Tokenization:** Text data was tokenized to convert words into numerical representations. This involves creating a vocabulary of unique words present in the dataset and mapping each word to an integer index.
* **Model Compilation:** The model was compiled with appropriate hyperparameters, including the choice of loss function, optimizer, and evaluation metrics. For binary classification tasks like text authorship detection, binary cross-entropy loss and Adam optimizer are commonly used choices.
* **Training:** The model is trained on the training data using the fit () method. During training, the model iteratively adjusts its parameters (weights and biases) based on the gradients of the loss function with respect to these parameters. The training process involves multiple epochs, where each epoch corresponds to one pass through the entire training dataset.
* **Validation:** Model performance is monitored on the validation set after each epoch to assess its generalization ability and prevent overfitting.
* **Early Stopping:** To prevent overfitting, early stopping can be employed, where training was halted if the performance on the validation set does not improve for a certain number of epochs.
* **Hyperparameter Tuning:** Hyperparameters such as learning rate, batch size, and number of layers can significantly impact the model's performance. These hyperparameters are tuned using techniques like grid search or random search to find the optimal configuration.
* **Evaluation:** Once training is complete, the final performance of the trained model is evaluated on the test set, which provides an unbiased estimate of its performance on unseen data.
* **Model Saving:** Finally, if the trained model meets the desired performance criteria, it can be saved to disk for future use or deployment in production environments.

By following these steps, we can train a CNN model for text authorship detection that achieves high accuracy and generalization ability on unseen data. Iterative refinement of the model architecture and hyperparameters may be necessary to achieve optimal performance.

**Sample Run**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generatedA screenshot of a text origin prediction

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

Top of Form

Bottom of Form

**Limitations:**

I used epoch = 4 which is very low compared to epoch = 10 at least. Due to the large dataset and limited computing power, it wasn’t possible to compute using more layers. Additional computing power and larger dataset can enhance the quality of the model and accuracy of predictions.