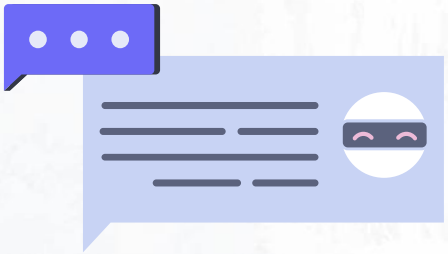
On Automating the Legal System : An AI-based Decision Support System for Judgements Prediction

(AI)



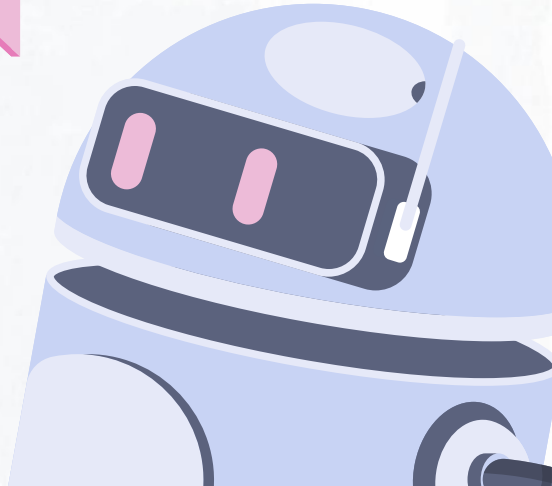


Table of contents

01 → Problem

02 → Motivation

03 → Objective

04 → Proposed Solution

05 → Dataset & Pre-processing

06 → Implementation

07 → Testing & Evaluation

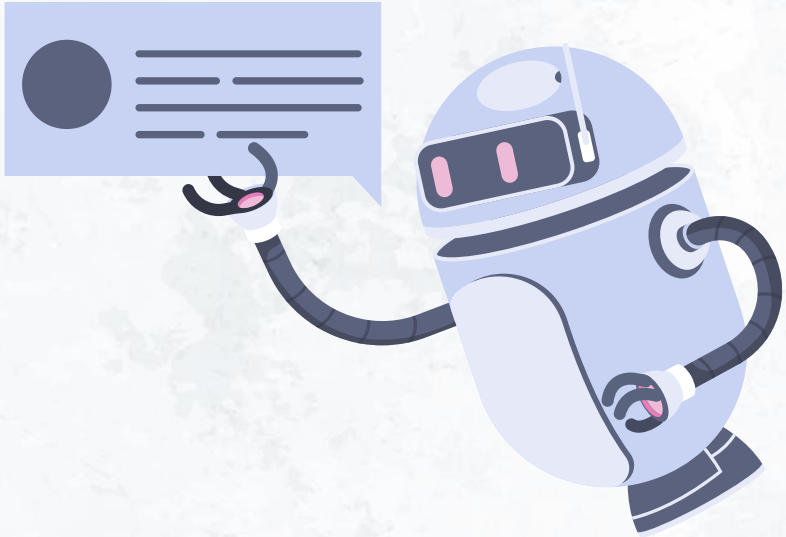
08 → Future Work

01 →

Problem

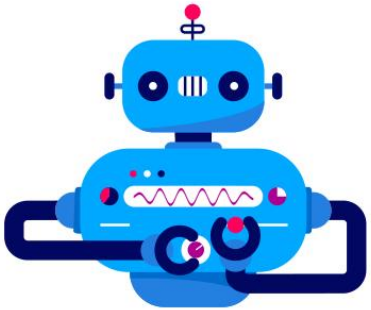
(AI)

In order to assist judges in their decision-making process, it is imperative to incorporate AI technology into the field of law. However, due to the sensitive nature of this domain, placing trust in an AI model for such crucial decisions can be challenging. Therefore, our aim was to develop a highly reliable model that is grounded in real data sourced from court proceedings and actual judicial rulings.



02 →

Motivation



(AI)

The number of cases has increased significantly compared to the number of judges.

- India: almost 24M court cases are pending for 17K judges.
- Brazil: over 1.6M court cases pending for department of 5 judges.
- US: 1.3M immigration cases for 500 judges Cases can take years if not decades to be heard

Difficulties:

1. - The availability of well-gathered English datasets is limited.
2. - We face tough competition from the top-ranked paper in the world on paper with code.
3. - The text length in the dataset is exceptionally large.
4. - BERT has limitations when it comes to processing long documents.

   What Makes BERT Stand Out? (BERT has showcased exceptional performance in various natural language processing tasks)

1. - The published paper lacks implementation details.

03 →

Objective

(AI)

The primary objective of this project is to develop a neural legal judgment prediction system using the ECHR dataset.

Our Main Task: binary violation classification

Additional Tasks:

1- case importance classification

2- multi-label classification.

By achieving these objectives, we seek to enhance the understanding and prediction of legal judgments, enabling more informed decision-making processes.

04 →

Proposed Solution

Methodology

Our methodology involves adapting the BERT model to handle long documents by employing truncation approaches and pooling techniques. We train and evaluate the adapted models using appropriate loss functions and evaluation metrics. We perform hyperparameter tuning and address class imbalance in the datasets. We analyze the performance of the models and compare them with existing literature. We also explore further enhancements such as incorporating larger transformer architectures. Through this methodology, we aim to overcome the limitations of BERT and achieve accurate classification and analysis of legal text data.

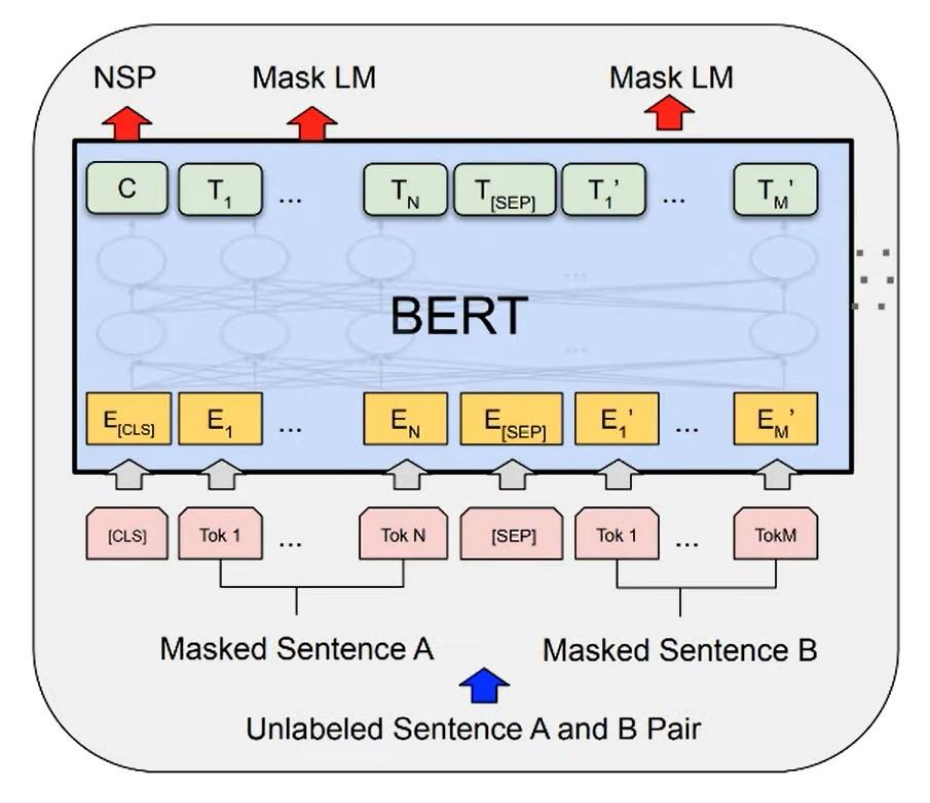
Steps

1. Conduct a comprehensive review of relevant literature and research papers on BERT.
2. We pre-process the ECHR dataset.
3. fine-tune the pre-trained BERT model to capture the semantic and contextual nuances of legal texts.
4. We design task-specific neural network architectures, incorporating BERT embeddings, to address the binary violation classification, case importance classification, and multi-label classification tasks.
5. Model Training and Evaluation.
6. Results Analysis and Comparison.
7. Conclusion and Future Directions.

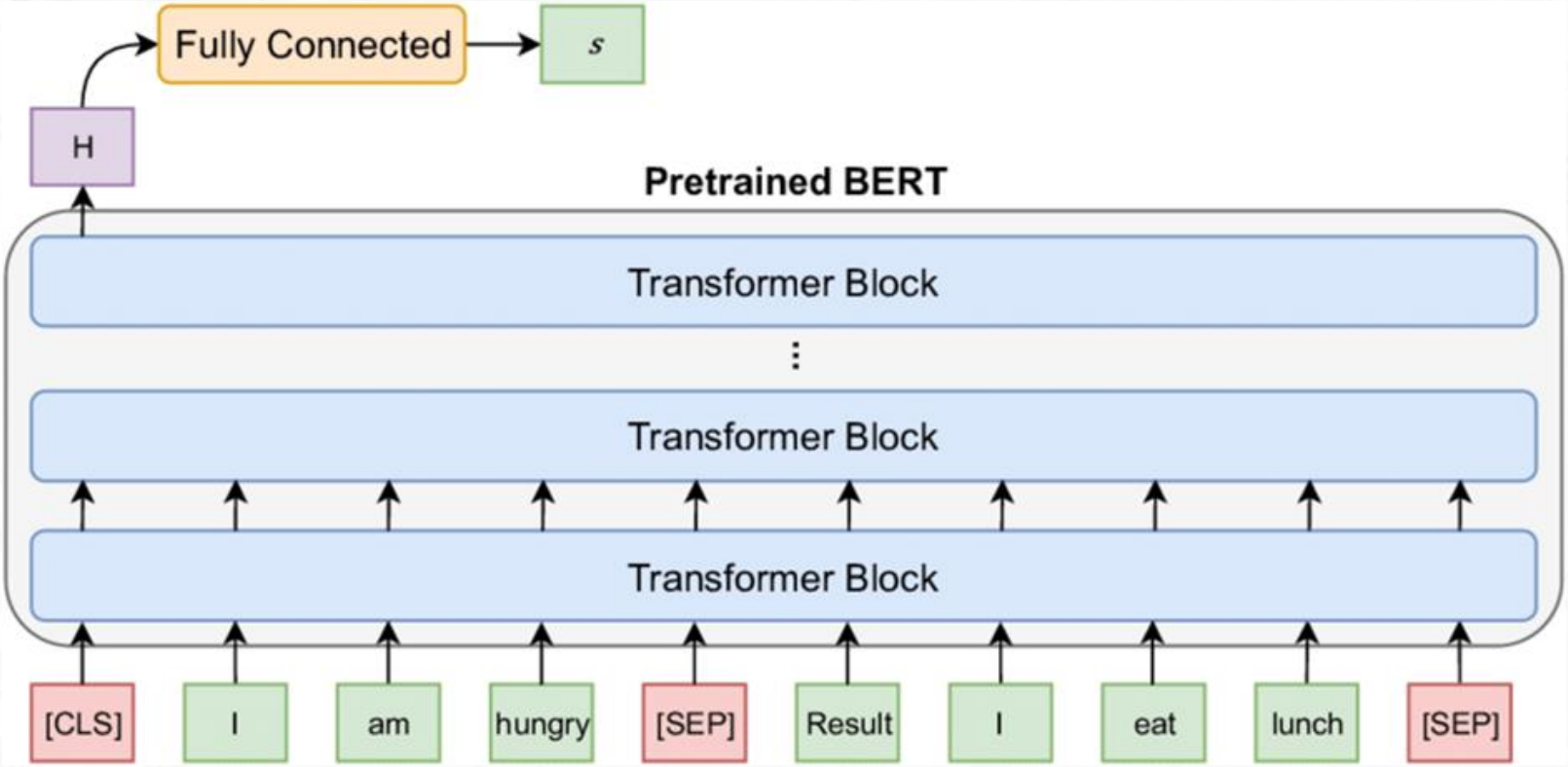
By combining the concepts and techniques from founded papers, we developed a unique approach that addresses the specific task at hand. We carefully analysed the methods proposed in each paper, identified their strengths and weaknesses, and integrated them to form a cohesive solution.

Bert Architecture

We have decided to choose **BERT** Architecture because we found that is the suitable architecture for our problem and also fits our computational resources.

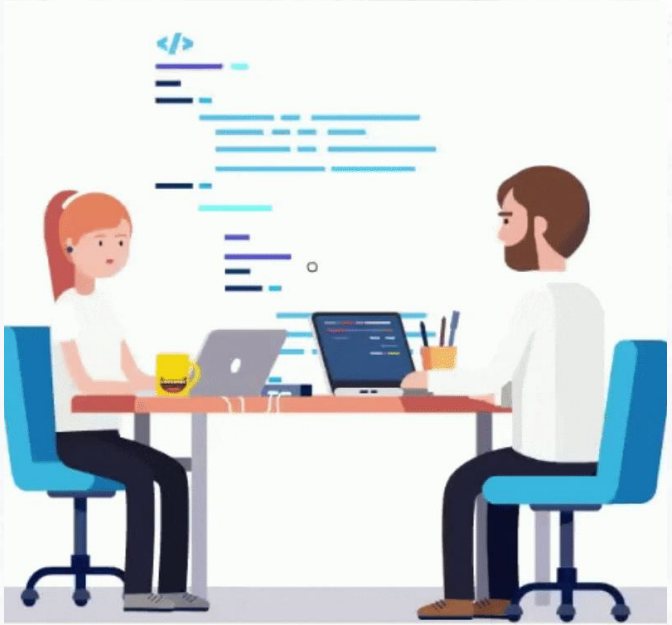


Bert Architecture



Functional Requirements

1. Data Wrangling
2. Pre-Processing and Tokenization
3. Truncation Methods
4. Hierarchical Methods
5. Binary Violation Classification
6. Case Importance Prediction
7. Multi-label Classification
8. Integration of LSTM and Transformer Encoder



Non-Functional Requirements

1. **Performance Evaluation:** The system should provide mechanisms for evaluating the performance of different neural models and determining the effectiveness and efficiency of the developed solution.

2. **Scalability and Compatibility:** The system should be scalable to handle increasing volumes of legal judgment data and compatible with the Nvidia A100 and T4 GPUs for efficient model training.

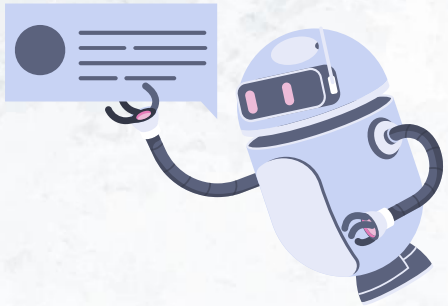
05 →

Dataset
& Pre-processing

(AI)

European Court of Human Rights (ECHR)

we utilized the European Court of Human Rights (ECHR) dataset, The dataset is provided in the form of JSON files.



The dataset is divided into three parts:

1. the training set -> 7100
2. validation set -> 1380
3. test set -> 2998

However, after hyperparameter tuning, we concatenate the training and validation sets into a single set, which is then used as the training data.

“This is a common approach to utilize all available labelled data for training after optimizing the hyperparameters.”

How we pre-processed the data?

Firstly, we convert the data from JSON format to CSV format. The ECHR dataset we utilized in our project consists of 16 columns. However, we focused on three specific columns that were relevant to our task. The first column is the “Text" column, which contains the real information of each legal case. The second column is the "Importance" column, which assigns a numerical value ranging from 1 to 4 representing the importance of each case. The third column is the “VIOA” (Violated articles) column, indicating the articles that were violated in each case.

Secondly, From the "Violated articles" column, we derived additional columns for our classification tasks. The first one is the "Binary violation" column, where each sample is assigned a value of 1 if one or more articles were violated, and 0 otherwise. This column was used for the binary classification task.

Finally, The rest derived column is a set of 14 columns, one for each article mentioned in the "Violated articles" column. Each of these columns contains a binary value, where 1 indicates the presence of the corresponding violated article, and 0 indicates its absence. These columns were utilized for the multilabel classification task, allowing us to predict the presence or absence of specific violated articles.

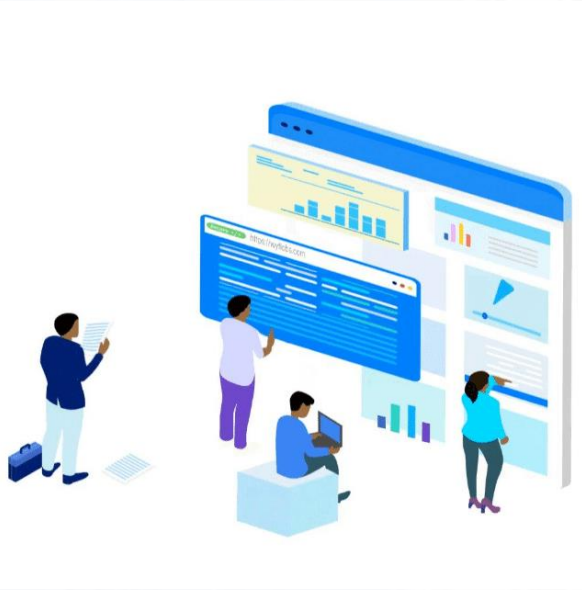
By leveraging these columns and their derived counterparts, we were able to construct appropriate datasets for our binary violation, Case Importance classification and multilabel classification tasks, facilitating the training and evaluation of our models.

Also we observed during our analysis that the dataset contains very large texts, with the maximum text length reaching 51334 tokens. Processing such long texts would require significant computational resources, especially when considering **All-truncation** approach (we will talk about later). Therefore, to manage computational complexity for this approach, we decided to focus on texts with a length smaller than **15000** tokens. This allows us to effectively train and evaluate the models while reducing the computational burden.

Note : For multilabel classification, we selected the labels of violated articles that had a relatively high frequency. This approach was inspired by our main paper.

06 →

Implementation

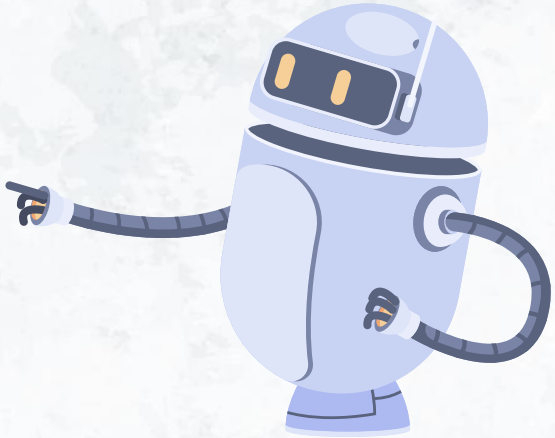


(AI)

We implement 7 approaches:

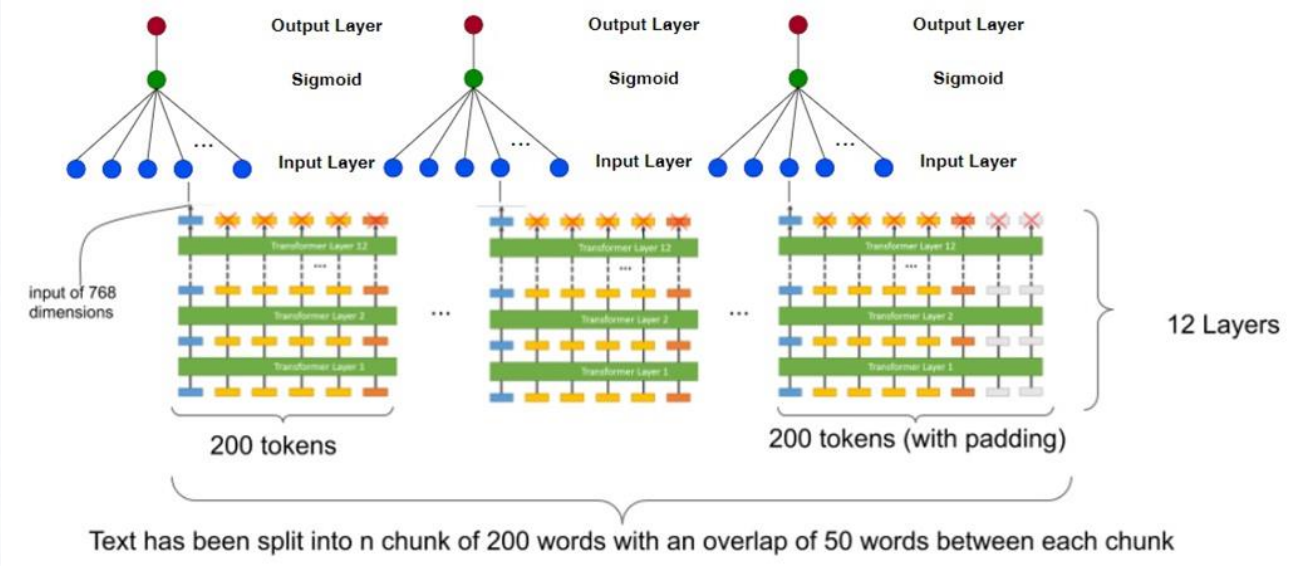
1. All-Truncation approach
2. Head-Truncation approach
3. Tail-Truncation approach
4. Mean-Hierarchical approach
5. Max-Hierarchical approach
6. Ro-BERT
7. To-BERT

let’s talk about them



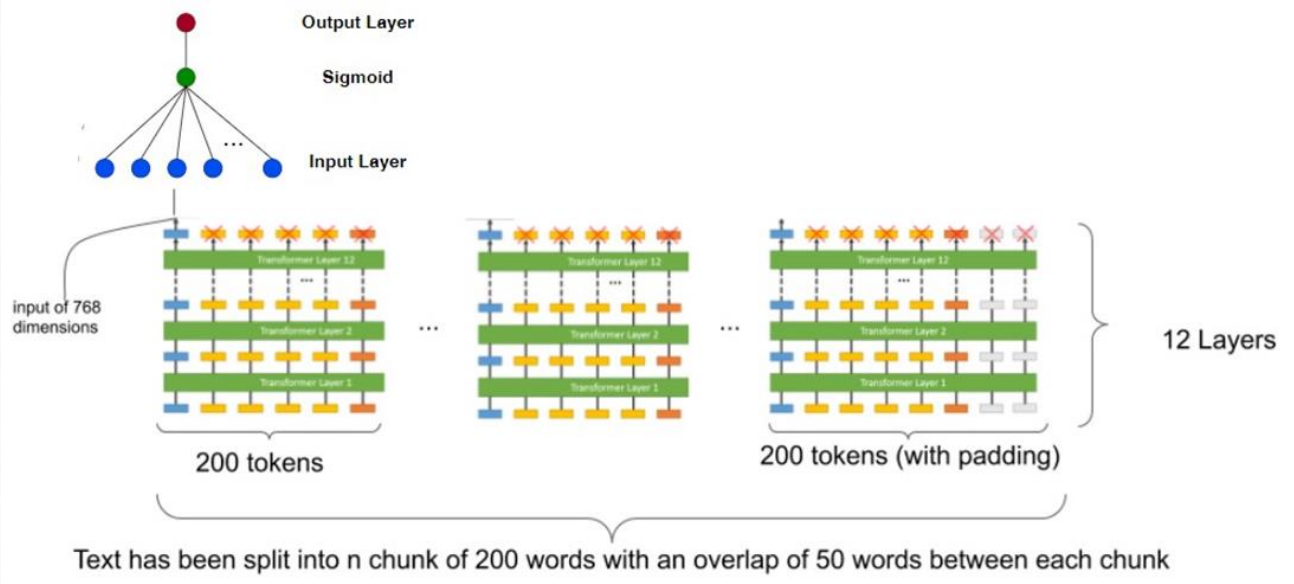
1- All-Truncation Approach

In this approach, we will consider each chunk of 200 tokens as a new document, so if a document is split into 3 chunks of 200, tokens we will consider each chunk as a new document with the same label.



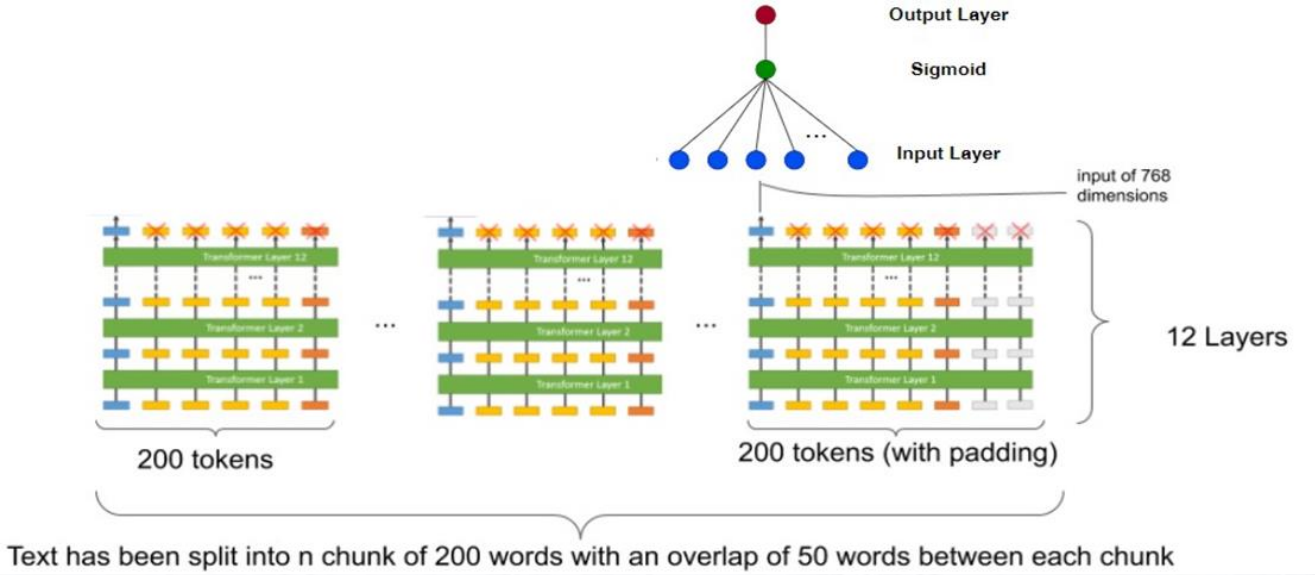
2- Head-Truncation Approach

In this approach, we will keep only the first chunk of 200 tokens for each document, so if a document is split into 3 chunks of 200, we will only keep the first chunk and we will not keep the last two chunks.



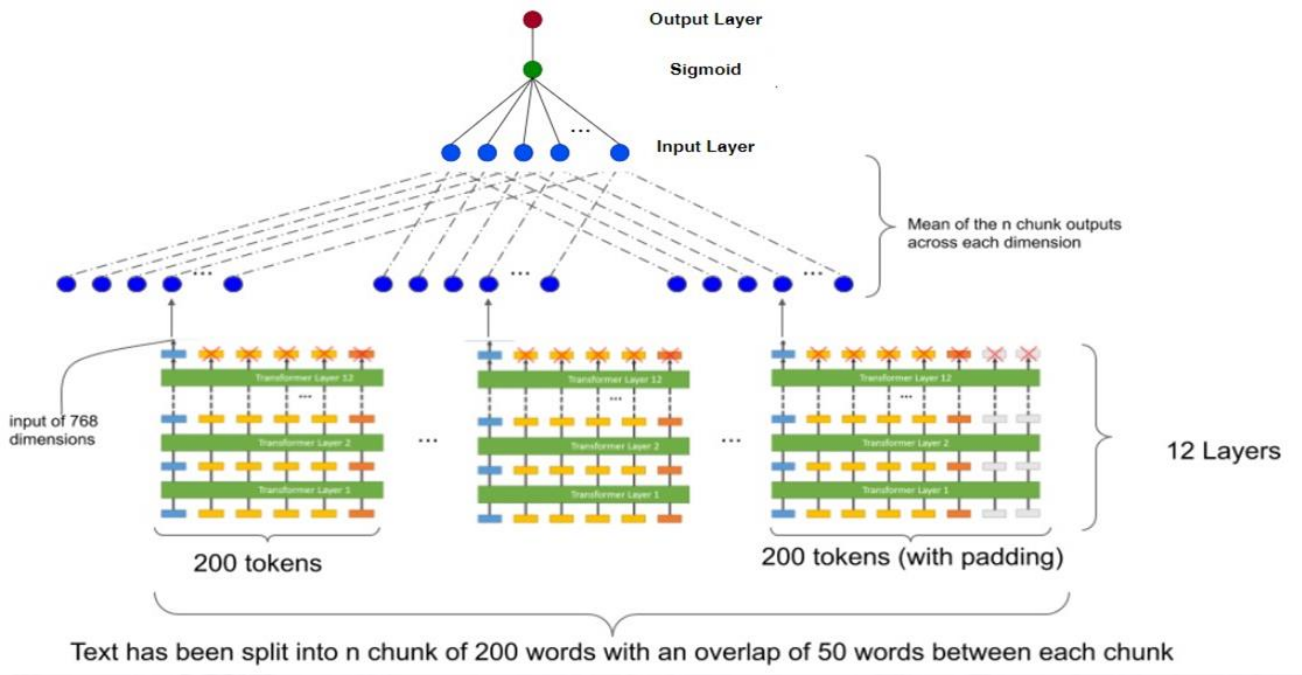
3- Tail-Truncation Approach

In this approach, we will keep only the last chunk of 200 tokens for each document, so if a document is split into 3 chunks of 200, we will only keep the last chunk and we will not keep the first two chunks.



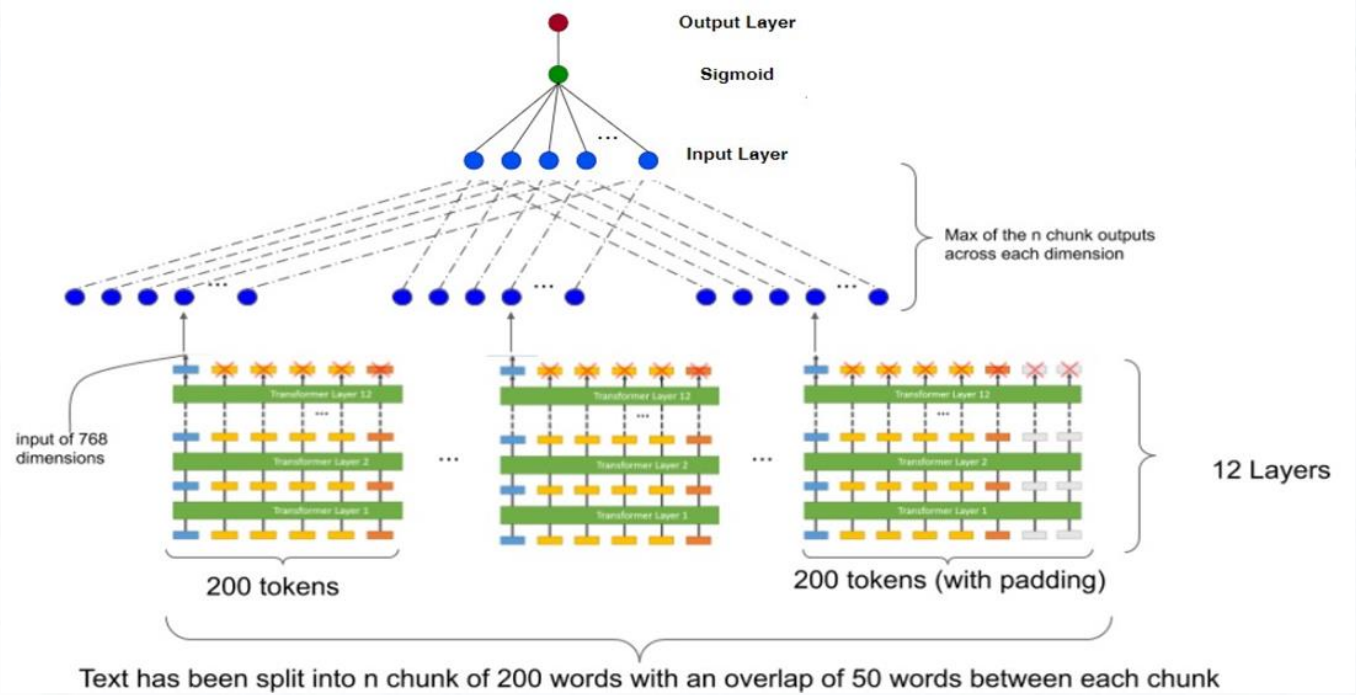
4- Mean-Hierarchical approach

in this approach, we average the embedding of all N chunks across each dimension.



5- Max-Hierarchical approach

in this approach, we take the maximum embedding of all the N chunks across each dimension.

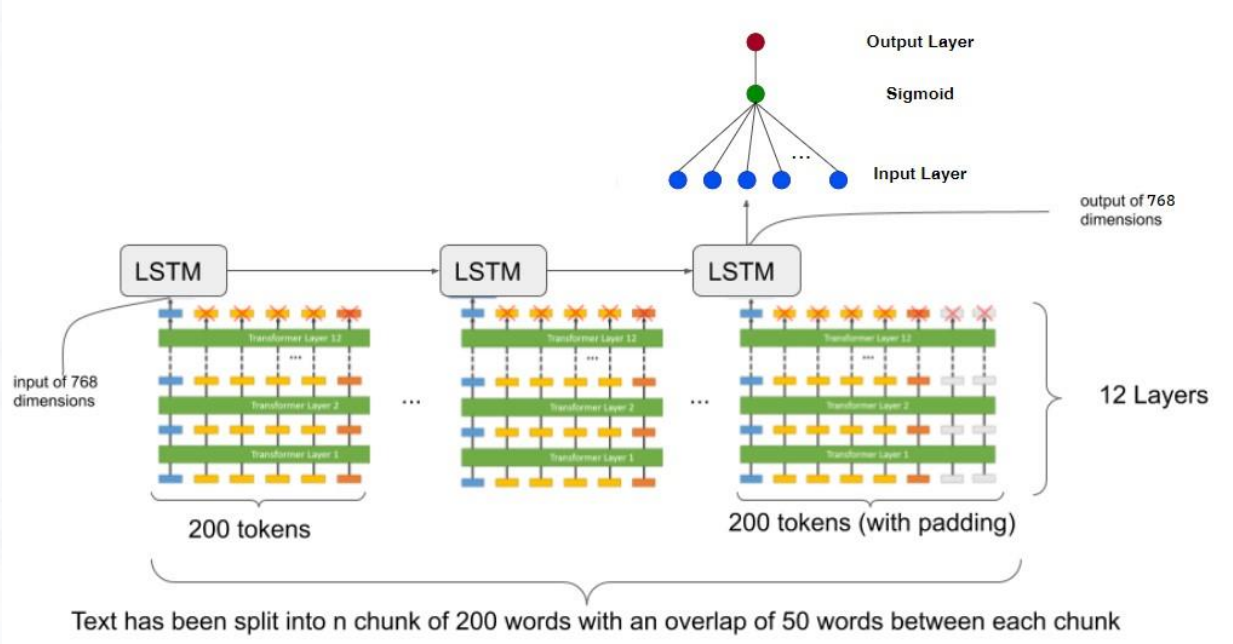


6- Ro-BERT

In this approach, the input text is divided into multiple chunks, which are then processed using a fine-tuned BERT model.

To preserve the order and remove the limitation of document length, we feed each chunk embedding into an LSTM cell. The dynamic nature of LSTM allows for accommodating documents of various sizes.

Finally, we pass the last hidden state from the LSTM to a neural network.

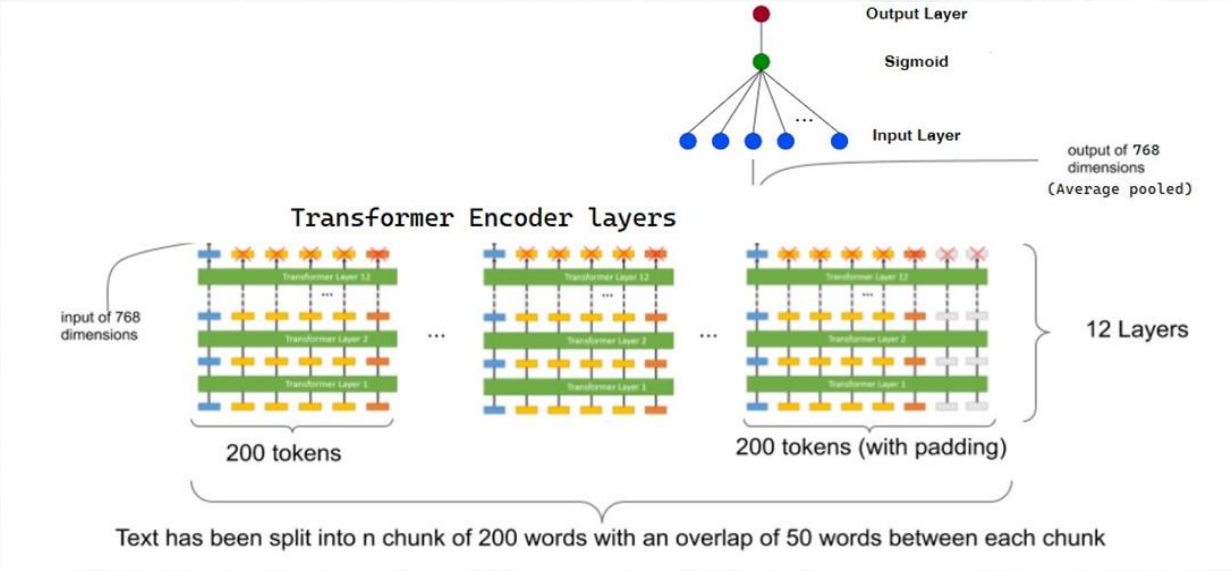


7- To-BERT

Like Ro-BERT, the input text is divided into n chunks, which are fed into the fine-tuned BERT. However, instead of using an LSTM layer, we utilized a transformer encoder to process the chunk embeddings.

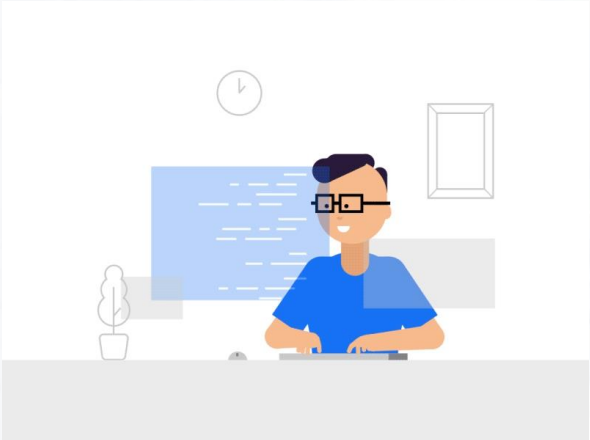
The transformer encoder captures the contextual information and dependencies between the tokens in a more efficient and parallelizable manner. It allows for better modelling of global dependencies

Finally, we passed the last hidden state from the transformer encoder to a neural network.



07 →

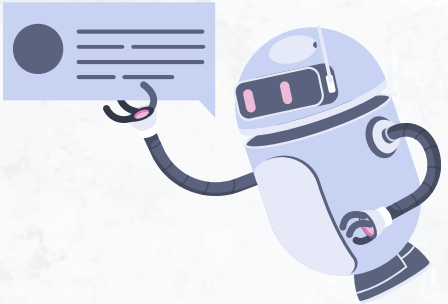
Testing & Evaluation



Testing & Evaluation

The evaluation of various approaches for **binary violation classification**. The **mean pooling** hierarchical approach achieved the highest performance with a **macro F1 score of 83.41%**, followed closely by the **Ro-BERT model with a score of 83.32%**. The **max pooling** hierarchical approach attained a **macro F1 score of 83.28%**. Our approach achieved remarkable results, surpassing the previous state-of-the-art model's **macro F1 score of 82%** which is at the top rank in the Papers with Code repository. For more details, you can check the following link:

Neural Legal Judgment Prediction in English | Papers With Code.



Testing & Evaluation

All-truncation approach achieved an **F1 macro score of 76.95%**. This outcome was expected because treating each chunk as a separate document could include irrelevant words that do not contribute to the key facts of the case.

Head truncation approach resulted in a **macro F1 score of 80.31%**, indicating that important key facts are primarily located at the beginning of the text.

Tail truncation approach achieved a **macro F1 score of 78.85%**, suggesting that crucial information is concentrated towards the end of the text.

To-BERT approach achieved a **macro F1 score of 80.71%**, indicating its potential for further improvements and modifications.

Binary Violation Classification Results

Model / Metric	*Macro F1 Score*	*Macro Precision*	*Macro Recall*
All	76.95 %	80.31 %	75.47 %
Head	80.31 %	85.12 %	78.34 %
Tail	78.85 %	85.58 %	76.65 %
Mean pooling	**83.41 %**	90.85 %	80.76 %
Max pooling	83.28 %	90.48 %	**80.86 %**
Ro-BERT	83.32 %	**91.29 %**	80.59 %
To-BERT	80.71 %	87.74 %	78.33 %

Table 1: Binary Violation Results

Case Importance Classification Results

Model / Metric	Micro F1 Score	Micro Precision	Micro Recall
Mean pooling	74.47 %	74.47 %	74.47 %

Table 2: Case Importance Results

We also tackled the task of case importance as a regression problem, as outlined in our main paper. We aimed to predict the importance of each case on a numerical scale. After thorough experimentation and analysis, we were able to surpass the results reported in the paper by a small improvement of mean absolute error value with 0.067.

Multi-label Classification Results

Model / Metric	Micro F1 score	Micro Precision	Micro Recall
Mean pooling	30.13 %	18.64 %	78.55 %

Table 3: Multi-Label Classification Results

We employed ensemble methods to combine the outputs of binary classification models in two ways: **majority voting** and **averaging sigmoid values**. Both methods produced very similar results, indicating their effectiveness in combining multiple models.

The best combination from majority voting was determined based on the Macro F1 score is ('tobert', ' & ', 'tail', ' & ', 'max', ' & ', 'robert', ' & ', 'head'), resulting in the selection of ‘0.835%’ as the optimal approach.

The best combination from averaging sigmoid values was determined based on the Macro F1 score is ('tobert', ' & ', 'tail', ' & ', 'max'), resulting in the selection of ‘0.836%’ as the optimal approach.

Despite our efforts to apply **ensemble methods** to improve the results, we found that they did not lead to significant improvements in our specific scenario.

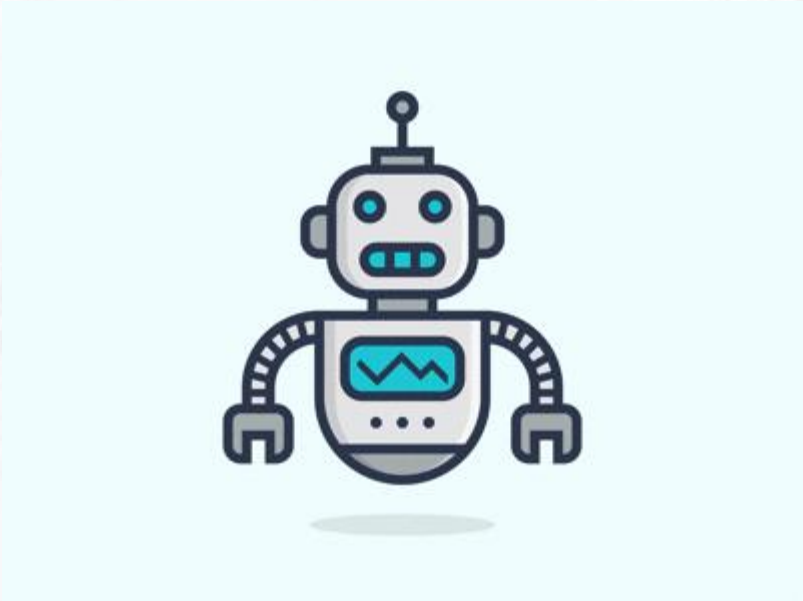
There are a few possible reasons for this outcome:

1 - **Lack of Diversity:** Ensemble methods thrive on the diversity of the base models. If the different approaches you used with BERT are not diverse enough, the ensemble may not be able to capture a wider range of patterns in the data. In such cases, the individual models might have similar errors or make similar mistakes, leading to limited improvement when combined.

2 - **Inherent Limitations:** It's important to acknowledge that ensemble methods might not always lead to improved performance, especially if the data or the problem itself has inherent limitations. In some cases, the ceiling of performance might have already been reached, and no further improvement is possible.

08 →

Future Work



(AI)

We could enhance our work by incorporating larger transformer models such as **GPT** or **XLNet**. However, it's crucial to consider the potential challenges in terms of computational resources. These larger models typically require more memory and computational power for training and inference. Therefore, it is essential to have sufficient resources or explore strategies to handle

these computational demands, such as distributed training or utilizing cloud-based infrastructure. Balancing the benefits of using bigger models with the practical constraints of computational resources is an important consideration for further improvement in our work

Thanks!