# Automatic Rhetorical Roles Classification for Legal Documents using LEGAL-TransformerOverBERT

Gabriele Marino, Daniele Licari*, Praveen Bushipaka, Giovanni Comandé and Tommaso Cucinotta

*Scuola Superiore Sant'Anna, P.zza dei Martiri della Libertà, Pisa, 56100, Italy*

Abstract

Automatic identification of rhetorical roles can help in many downstream applications of legal documents analysis, such as legal decisions summarization and legal search. This is usually a complex task, even for humans, due to its inherent subjectivity and to the difficulty of capturing sentence context in very long legal documents. We propose a novel approach, based on Hierarchical Transformers, which overcomes these problems and achieves promising results on two different datasets of Italian and English legal judgments. Specifically, we introduce LEGAL-TransformerOverBERT (LEGAL-ToBERT), a model based on the stacking of a transformer encoder over a legal-domain-specific BERT model, and show that our approach is able to significantly improve the baselines set by the stand-alone LEGAL-BERT models, by capturing the relationships between different sentences of the same document. We make our models available and ready-to-use for downstream applications of rhetorical roles classification in the legal context both for the Italian and English language.

**Keywords**

Rhetorical Roles Classification, LEGAL-BERT, Hierarchical Transformers, LEGAL-ToBERT

## 1. Introduction

Rhetorical Roles Classification (RRC) is a Natural Language Processing (NLP) task that aims to classify the semantic function of the sentences of a text. When working on new legal cases, legal practitioners often need to retrieve all the preceding relatable court decisions and from them extract the relevant information for their specific legal case, such as the determining facts and principles of those past decisions. In legal documents, sentences are strategically constructed to serve specific rhetorical purposes, such as asserting, providing evidence or examples, refuting a counterargument, or concluding an argument. The task of extracting this information from old cases is not only time-consuming, but often subject to ambiguity and difficult for even human experts. An automated tool is then crucial to save time and effort and speed up legal practitioners' work.

Many works have already shown how automatic RRC in legal texts can lead to enormous benefits for applications such as summarization, question answering, case analysis, argument extraction, judgment prediction and so on [1, 2, 3, 4, 5]. Previous approaches to RRC have relied on traditional machine learning algorithms such as Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and Long-Short Term Memory (LSTM) [6, 7, 8]. More recently, transformer-based language models pre-trained on a large corpus of legal texts have achieved significant, state-of-the-art results in many legal NLP tasks [9, 10, 11, 12]. The main challenge that transformer models have to face when dealing with RRC in the legal context is dictated by the length of legal documents, which makes it difficult to take into account relationships between sentences.

To address this issue we propose a novel model, that we named LEGAL-TransformerOverBERT (LEGAL-ToBERT). Our approach is based on the stacking of a transformer encoder on top of a legal-domain-specific BERT model, creating a hierarchical architecture able to capture the discursive relationships between sentences, allowing accurate classification of rhetorical roles. We also propose a novel positional encoding strategy for the upper-layer transformer of ToBERT, based on the sinusoidal encoding of the relative position of a sentence in the document, and show that this is preferable when dealing with RRC in the legal context.

As a proof of the effectiveness of our approach, we tested our model using two different datasets. The first one is a new yet confidential Italian-language dataset that we built specifically for this task and named ITA-RhetRoles; the second one is the English-language BUILD benchmark dataset [5]. We used respectively Italian-LEGAL-BERT [9] and LEGAL-BERT [12] as building blocks for LEGAL-ToBERT. We then compared the re-

*Proceedings of the Sixth Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2023), June 23, 2023, Braga, Portugal.*

gabriele.marino@santannapisa.it (G. Marino);
daniele.licari@santannapisa.it (D. Licari);
praveen.bushipaka@santannapisa.it (P. Bushipaka);
giovanni.comande@santannapisa.it (G. Comandé);
tommaso.cucinotta@santannapisa.it (T. Cucinotta)
0009-0005-2280-4637 (G. Marino); 0000-0002-2963-9233 (D. Licari); 0009-0009-7753-8662 (P. Bushipaka); 0000-0003-2012-7415 (G. Comandé); 0000-0002-0362-0657 (T. Cucinotta)

sults of LEGAL-ToBERT with those of the stand-alone Italian-LEGAL-BERT and LEGAL-BERT, and found that LEGAL-ToBERT allows for significantly better performances on both datasets, improving the baseline MCC respectively by 21% and 30%.

We make all our code and models publicly available and ready-to-use for downstream applications of legal RRC on our Rhetorical Roles Classification GitHub repository[1].

## 2. Related Work

In spite of the increasing research in applications of Artificial Intelligence to the legal domain, only limited works have focused on RRC. One of the earliest works with this aim can be traced back to Hachey et al. [13], in which handmade annotated sentences were used to train traditional Machine Learning algorithms such as Naive Bayes and SVM. Moens et al. [14] used Multinomial Naive Bayes classifiers and Maximum Entropy models to address the problem of argument detection in legal texts, as a particular case of RRC. Saravanan et al. [15] employed Conditional Random Fields (CRF) to automatize the RRC of legal documents and used the predicted rhetorical roles to rank each sentence and enable a subsequent extractive summarization task. More recently, a work by Ghosh et al. [8] used Hierarchical BiLSTM classifiers with the addition of a CRF to improve the stand-alone CRF baseline for RRC of Indian legal judgments. Starting from the results of this work, Malik et al. [16] proposed a Multi Task Learning (MTL) framework based on the same Hierarchical BiLSTM with CRF model to significantly improve the classification scores. Another noteworthy work by Walker et al. [6] investigated the use of ML and rule-based approaches for RRC tasks, and interestingly found that both approaches can lead to very promising results with a small dataset of manually labeled sentences.

With the advent of deep learning and transformer models [17], neural methods have been applied to RRC, significantly improving the results with respect to previous works. Bhattacharya et al. [18] experimented on cross-jurisdictional legal documents datasets with various models including Hierarchical BiLSTM and GRU with the addition of a CRF and with the integration of an attention mechanism. They compared these models with LEGAL-BERT [12], a legal-domain-specific pre-trained transformer, which outperformed the other traditional machine learning algorithms, suggesting to investigate further in the direction of transformers applications to RRC in the legal domain.

Some other works have shown how hierarchical transformers architectures can be employed to improve the performances of standard transformers when dealing with long texts [19, 20, 21].

Our experiments address RRC using a hierarchical transformer architecture based on legal-domain-specific BERT models. To the best of our knowledge, this is the first attempt combining these two colliding worlds and using them to build a refined model for RRC in the legal domain. Our models are available and ready-to-use both for the Italian and English language. This is the first time that a fine-tuned model is made available for RRC of legal documents for the Italian language: it is our sincere hope that this will enable many downstream applications, helping to speed up the work of Italian jurists.

## 3. Methodology

### 3.1. Rhetorical Roles Datasets

We used two different datasets to compare the performances of our hierarchical model with those of vanillla BERT models. The first one is a novel dataset that we developed for this work and that we named ITA-RhetRoles, the second one is the BUILD benchmark dataset [5]. Table 1 shows an overview of the two datasets in terms of number of documents and total sentences; both datasets are described more in details in the following sections.

| Split | ITA-RhetRoles | | BUILD | |
|---|---|---|---|---|
| | #Docs | #Sents | #Docs | #Sents |
| Train | 1045 | 68,012 | 221 | 25,752 |
| Valid. | 149 | 9,620 | 24 | 3,234 |
| Test | 294 | 18,288 | 30 | 2,879 |

**Table 1**
Number of documents and total number of sentences for ITA-RhetRoles and BUILD datasets.

#### 3.1.1. ITA-RhetRoles Dataset

ITA-RhetRoles is a dataset of civil law Italian legal cases. This dataset has been kept private as it was built under a confidentiality agreement between Scuola Superiore Sant'Anna and some Italian courts. ITA-RhetRoles consists of approximately 1,500 Italian legal documents, split into train, validation, and test set using the year and the subject of the case as stratification keys. Figure 1 shows the dataset distribution in terms of documents length: the longest document of the dataset consists of 248 sentences. The labelled rhetorical roles are the 5 most common sections of an Italian civil judgment: "Introduction" (INT), "Conclusions of the parties" (CP), "Summary of the appealed judgment" (SAJ), "Legal reasons" (LR), and "Decisional content" (DC). These labels were extracted using regular expressions to identify the different sections in the collected documents. Handmade validation was then

[1] https://github.com/GM862001/RhetoricalRolesClassification

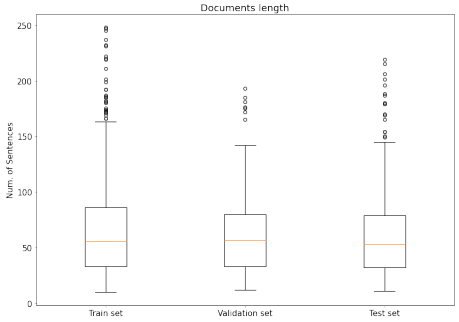


**Figure 1:** Distribution of documents lenght for ITA-RhetRoles dataset.

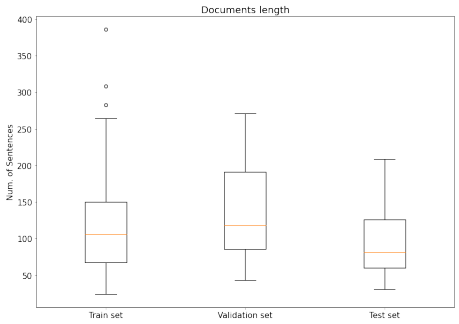


**Figure 2:** Distribution of documents length for BUILD dataset.

performed on a significant subset of the documents to assess the quality of the dataset.

#### 3.1.2. BUILD Dataset

BUILD dataset is a corpus of legal judgment documents from the Supreme Court of India, High Courts in different Indian states and some district-level courts. It consists of a publicly released train and validation set[2] and a private test set. We used the public validation set as test set and split the original train set into a train and validation set. Figure 2 shows the dataset distribution in terms of documents length: the longest document of the dataset consists of 386 sentences. The labelled rhetorical roles are 13: "Preamble" (PRE), "Facts" (FAC), "Ruling by Lower Court" (RLC), "Issues" (ISSUE), "Argument by petitioner" (ARGP), "Argument by respondent" (ARGR), "Analysis" (ANA), "Statute" (STA), "Precedent relied" (PRER), "Precedent not relied" (PRENR), "Ratio of the decision" (RAT), "Ruling by Present Court" (RPC), "None of the others" (NONE).

[2]https://github.com/Legal-NLP-EkStep/rhetorical-role-baseline

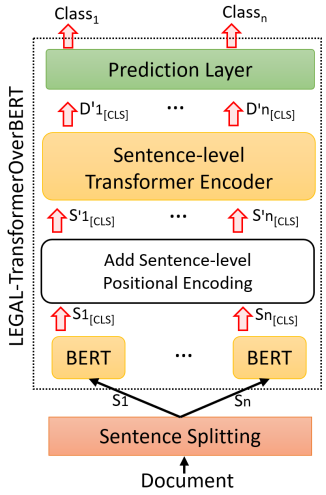


**Figure 3:** LEGAL-ToBERT model architecture.

### 3.2. TransformerOverBERT (ToBERT)

TransformerOverBERT (ToBERT) has a hierarchical architecture, shown in figure 3, based on the stacking of the following components: a BERT token-level encoder, a sentence-level positional encoder, a sentence-level encoder, and a prediction layer. The processing of a legal case starts with splitting the raw text of the document into sentences and tokenizing them. Each sentence is fed to the BERT token-level encoder and the pooled output for that sentence, i.e. the hidden representation of the [CLS] token output by BERT, is extracted.

The pooled outputs are gathered and fed to the positional layer to create a position-dependent encoding of each sentence in the document. These are then input into the sentence-level encoder. The output representations of this layer are finally fed to the prediction layer for rhetorical roles classification. Each of these components is described in the following sections.

#### 3.2.1. Data Preprocessing

Before being input to ToBERT, each document is split into sentences. Each sentence is tokenized using the token-level BERT tokenizer and then padded or truncated to a certain number $T$ of tokens. Documents are also padded with null sentences up to the length of the longest document of the train set (e.g., 386 for BUILD dataset and 284 for ITA-RhetRoles dataset), so to have a batch of input documents $\mathcal{J} \in \mathbb{R}^{D \times S \times T \times E}$, where $D$ is the number of documents, $S$ is the number of sentences for each document, and $E$ is the size of the token embeddings.

#### 3.2.2. BERT Token-Level Encoder

Bidirectional Encoder Representations from Transformers (BERT) is a neural model based on the transformer architecture [17]. It uses self-attention, residual connections, and layer normalization to achieve state-of-the-art results in many different tasks, with the addition of a task-specific output layer as the only modification to the model architecture [22].

BERT-like models are usually pre-trained via self-supervised methods on large unlabelled corpora and then fine-tuned for the specific task in a supervised fashion. Our approach is not different, in that we leverage two different pre-trained BERT models: Italian-LEGAL-BERT [9] and LEGAL-BERT [12]. Both these models are pre-trained on huge legal datasets consisting of Italian and English cases respectively: our training process aimed only to fine-tune them for our RRC use case.

In ToBERT, BERT is used as a token-level encoder. Specifically, it is used to obtain the hidden token representation $[CLS]$ of each batch sequence. It means that it is fed with $D$ batches of sentences $\mathcal{S} \in \mathbb{R}^{S \times T \times E}$ and produces as output a set of $D$ document representations $\mathbb{R}^{S \times H}$, where $H$ is the hidden size of the specific BERT model used (e.g., 768 for LEGAL-BERT).

#### 3.2.3. Sentence-Level Positional Encoder

A specific positional encoder is used to add a piece of information to the representation of each sentence about its position in the document.

In this work, we focus on *Sinusoidal Positional Embeddings*. Let's define the input document length (i.e. the number of sentences in the document) as $S$ and the embedding dimension as $H$ (e.g., 768 for Legal-BERT). For the $t$-th sentence representation $s \in \mathbb{R}^H$ of a document (with $0 \leq t < S$), the output of a sinusoidal positional encoder is:

$$s' = s + p_t,$$

where the $i$-th component of the embedding vector $p_t \in \mathbb{R}^H$ (for $0 \leq i < H$) is given by:

$$p_t^i = \begin{cases} \sin(w_t \omega_k), & \text{if } i \text{ is even} \\ \cos(w_t \omega_k), & \text{if } i \text{ is odd} \end{cases}$$

where

$$\omega_k = \frac{1}{10000^{\frac{2i}{H}}}.$$

and $w_t$ are weights that depend on the embedding strategy.

We tried two different approaches. The first one, which we named *Absolute Positional Embedding*, is the same used in the original Transformer architecture [17], and uses the weights $w_t = w_t^a = t$. The second one is a novel embedding strategy that we named *Relative Positional Embedding*. This one takes into account the relative position of a sentence in the document and uses the weights $w_t = w_t^p = \frac{1000t}{d}$, where $d$ is the length of the document to which the sentence belongs[3]. Basically, instead of encoding the absolute position of a sentence, this embedding strategy encodes the relative position of that sentence with respect to the length of the document in thousandths (‰), using standard Sinusoidal Positional Embeddings. The idea behind this approach is that legal documents often have a repetitive rhetorical structure (introductory sentences always come first, followed by sentences summarizing the final decision, and so on). By a preliminary explorative data analysis we found that there exists in fact a correlation between the rhetorical role of a sentence and its relative position in the document. This dependency might rely on the specific language and legal field of the document, but for sure including such a piece of information to the positional encoding of a sentence might add valuable hints for its correct rhetorical role classification.

#### 3.2.4. Sentence-Level Encoder

The sentence-level encoder is a transformer model [17] with the same configuration of the transformer encoders of the token-level BERT encoder (768 hidden dimensions, 12 attention heads, GELU activation function, and so on), but with only 2 stacked encoder-layers. It is used to process the batch of document representations $\mathcal{D} \in \mathbb{R}^{D \times S \times H}$ output by the positional encoder. The output produced by this component has the same shape as its input and is a batch of document representations that takes into account the relationships between the sentences of each document. The advantage of using a transformer encoder over recurrent architectures like LSTMs is that of better capturing long-distance relationships between sentences, thanks to the multi-head attention mechanism. This algorithm involves four main steps:

1. *Input*: a document representation $D \in \mathbb{R}^{S \times H}$ where $S$ is the number of sentences of $D$ and $H$ is the model's hidden size.
2. *Linear transformations*: the attention function is applied in parallel using $n_h = 12$ attention heads. For each attention head $i$, $D$ is projected into three different spaces: the key space $K_i \in \mathbb{R}^{S \times d_k}$, the query space $Q_i \in \mathbb{R}^{S \times d_k}$, and the value space $V_i \in \mathbb{R}^{S \times d_v}$. These projections are computed using learned weight matrices $W_i^K \in \mathbb{R}^{H \times d_k}$, $W_i^Q \in \mathbb{R}^{H \times d_k}$, and $W_i^V \in \mathbb{R}^{H \times d_v}$, where $d_k = d_v = 64 (H / n_h)$.
3. *Scaled dot-product attention and softmax*: for each attention head $i$, the attention scores are computed by taking the dot product of query $Q_i$ and

[3] We do not take into account the padding sentences here.

key $K_i$, scaling by the square root of the key dimension $d_k$, and then applying a softmax function to normalize the scores. Finally, the normalized scores are multiplied by the value matrix $V_i$ to obtain the attention output matrix $O_i$:

$$O_i = \text{softmax}\left(\frac{Q_i \times K_i^\top}{\sqrt{d_k}}\right) V_i$$

4. *Output*: after computing the output $O_i$ for each attention head, these are concatenated along their last dimension. Finally, a linear projection is applied using a learned weight matrix $W_O \in \mathbb{R}^{n_h \times d_v \times H}$ to obtain the final output of the Multi-Head Attention layer.

$$O = \text{concat}(O_1, O_2, ..., O_{heads})W_O$$

This sentence-level multi-head attention mechanism allows the model to capture different types of relationships between sentences by learning separate attention patterns for each head. Instead, stacking multiple encoder layers allows to learn increasingly abstract representations of the input sequence.

Similar to the transformer encoders used in BERT architecture, this model includes a dropout layer as a regularization technique to prevent overfitting. Dropout randomly shuts down some of the neurons in the network during training, sampling from a Bernoulli distribution with some probability $p$ (which is equal to 0.1 in case of BERT), forcing the remaining neurons to learn more robust features that are not dependent on the presence of other units.

#### 3.2.5. Prediction Layer

The prediction layer input is the batch of document representations $\mathscr{D}' \in \mathbb{R}^{D \times S \times H}$ as output by the sentence-level encoder. This is fed to a linear layer with $n$ output units, $n$ being the number of labels (i.e. rhetorical roles), and then goes through a dropout layer for regularization purposes. The final output $\mathscr{O} \in \mathbb{R}^{D \times S \times n}$ is the rhetorical roles classification logits. If labels are provided (e.g. during training) this layer computes and returns the cross entropy loss between the logits and the labels, filtering out the inactive tokens (i.e. the padding ones).

## 4. Experiments

Our experiments aimed to provide a baseline for both ITA-RhetRoles and BUILD datasets using legal-domain-specific BERT models and improve them using LEGAL-ToBERT. When evaluating our models we considered the following metrics: accuracy, Matthew Correlation Coefficient (MCC), micro and macro precision, micro and macro recall, micro and macro F1.

### 4.1. Models

We used Italian-LEGAL-BERT [9] and LEGAL-BERT [12] (the baselines models) to provide a baseline respectively for ITA-RhetRoles and BUILD datasets. Specifically, each of them was chosen as the encoder of an `AutoModelForSequenceClassification` from `HuggingFace Transformers` Python package [23]. We coupled each model with the relative `AutoTokenizer`, and we applied truncation and padding using $T = 64$ as the max sentence length. As described in section 3.2.1, we also padded the documents with null sentences up to the length of the longest document for each dataset (386 sentences for BUILD dataset, 284 sentences for ITA-RhetRoles dataset).

After having set a baseline for both datasets, we used the very same BERT models as the token-level encoders of ToBERT, and used ToBERT itself as the encoder of an `AutoModelForTokenClassification`, keeping the same tokenizers and same truncation max sequence length. As sentence-level encoder we used 2 stacked encoder layers from `PyTorch` transformer model.

### 4.2. Training and Hyperparameters Fine-Tuning

We trained all our models using a `PyTorch` linear scheduler based on AdamW optimizer, leveraging the Gradient Scaler from the `CUDA Automatic Mixed Precision` package. When training the baseline models we set the batch size to 128, while we used one document batches to train ToBERT. In both cases, we accumulated gradients every 3 steps. We set a maximum number of epochs to 20, but contextually using early stopping with 2 patience steps. All other relevant hyperparameters were fine-tuned.

We used `Optuna` Python package for hyperparameters fine-tuning [24]. This is an automated and efficient optimization framework offering a versatile *define-by-run API* for the hyperparameters space.

When training our baseline models we considered the following hyperparameters space:

- Learning rate $\in [5e-6, 5e-4]$;
- Weight decay $\in [1e-3, 1e-1]$.

To these hyperparameters, we added the following ones when training ToBERT:

- Sentence-level positinal embedding strategy (S_lv_ pos_emb): either absolute or relative;
- Sentence-level encoder dropout (S_lv_enc_dropout) $\in [0.1, 0.7]$;
- Sentence-level encoder feed-forward network size (S_lv_enc_FFN_size) $\in 50, 51, ..., 1000$.

We used TPE (Tree-structured Parzen Estimator) algorithm proposed by Bergstra et al. [25] for hyperparameters optimization. This method has been shown to outperform many competiting ones, including random search and grid search, in terms of efficiency and effectiveness. By fitting two separate Gaussian Mixture Models (GMMs) to the best and worst objective values, TPE estimates the density of the promising and unpromising regions separately, and guides the search accordingly. On each trial, TPE samples a new set of candidate hyperparameters by maximizing the ratio $l(x)/g(x)$, where $l(x)$ is the density estimate of "good" hyperparameters combinations and $g(x)$ is the density estimate of "bad" hyperparameters combinations. The candidate hyperparameters with the highest ratio are then evaluated using the objective function, and the process is repeated.

For each dataset, we performed 32 search trials minimizing the validation loss, and picked the best model for final testing. Table 2 shows the best hyperparameters combination for LEGAL-BERT and LEGAL-ToBERT when trained on ITA-RhetRoles and BUILD datasets.

It is interesting to notice that in both cases the relative embedding strategy was prefered to the absolute one when training LEGAL-ToBERT. This suggests the effective usefulness of including relative position information in the positional embeddings of the sentences, to leverage the correlation between this feature and their rhetorical role, due to the repetitive rhetorical structure of a legal document as a whole.

| Dataset | Model | Parameter | Value |
|---|---|---|---|
| ITA-Rhet Roles | LEGAL-BERT | Learning rate | 6.49e-05 |
| | | Weight decay | 5.35e-02 |
| | LEGAL-ToBERT | Learning rate | 8.32e-05 |
| | | Weight decay | 6.93e-02 |
| | | S_lv_ pos_emb | relative |
| | | S_lv_enc_dropout | 0.26 |
| | | S_lv_enc_FFN_size | 167 |
| BUILD | LEGAL-BERT | Learning rate | 7.03e-05 |
| | | Weight decay | 9.16e-02 |
| | LEGAL-ToBERT | Learning rate | 7.54e-05 |
| | | Weight decay | 8.36e-02 |
| | | S_lv_ pos_emb | relative |
| | | S_lv_enc_dropout | 0.13 |
| | | S_lv_enc_FFN_size | 968 |

**Table 2**
Best LEGAL-BERT and LEGAL-ToBERT hyperparameters combination for ITA-RhetRoles and BUILD datasets.

# 5. Results

We evaluated our approach for legal RRC both on ITA-RhetRoles and BUILD dataset. Our analysis aims to compare the results of LEGAL-ToBERT with the baselines provided by vanilla stand-alone LEGAL-BERT models, both in overall terms and with respect to each considered rhetorical role.

## 5.1. ITA-RhetRoles

Table 3 lists the results of the best models selected by the hyperparameters fine-tuning process on the ITA-RhetRoles test dataset. LEGAL-ToBERT achieves almost perfect score in each considered metric (all of them always remain above 97%), significantly outperforming LEGAL-BERT. In particular, LEGAL-ToBERT achieves macro F1 score of 0.98 and MCC of 0.972, improving the baselines set by LEGAL-BERT by 12% and 21% respectively.

| Metric | | LEGAL-BERT | LEGAL-ToBERT |
|---|---|---|---|
| Accuracy | | 0.872 | **0.982** |
| MCC | | 0.806 | **0.972** |
| F1 | Macro | 0.878 | **0.980** |
| | Micro | 0.872 | **0.982** |
| P | Macro | 0.871 | **0.979** |
| | Micro | 0.872 | **0.982** |
| R | Macro | 0.889 | **0.980** |
| | Micro | 0.872 | **0.982** |

**Table 3**
Test results for ITA-RhetRoles dataset.

We also analyzed the performance of our method on each rhetorical role separately. Table 4 shows the precision, recall, and macro F1 score for each rhetorical role. In terms of macro F1 score, LEGAL-ToBERT achieves better performances for each rethorical role, apart from introductory sentences, for which the performances of the two models are comparable. Specifically, the improvement in terms of macro F1 scores ranges from 3% (DC - decisional sentences) to 16% (SAJ - sentences summarizing the appealed judgment).

| RR | LEGAL-BERT | | | LEGAL-ToBERT | | |
|---|---|---|---|---|---|---|
| | F1 | P | R | F1 | P | R |
| INT | 0.990 | 0.985 | 0.994 | **0.994** | **0.994** | **0.995** |
| CP | 0.896 | 0.866 | 0.935 | **0.990** | **0.986** | **0.993** |
| SAJ | 0.849 | 0.861 | 0.839 | **0.984** | **0.984** | **0.984** |
| LR | 0.912 | 0.912 | 0.912 | **0.986** | **0.986** | **0.986** |
| DC | 0.951 | 0.957 | 0.945 | **0.980** | **0.983** | **0.979** |

**Table 4**
Test results for each rhetorical role of ITA-RhetRoles dataset.

## 5.2. BUILD

Table 5 lists the results of the best models selected by the hyperparameters fine-tuning process on the BUILD test dataset. LEGAL-ToBERT significantly outperforms LEGAL-BERT in each considered metric. In particular,

LEGAL-ToBERT achieves macro F1 score of 0.57 and MCC of 0.73, improving the baselines set by LEGAL-BERT by 22% and 30% respectively.

| Metric | | LEGAL-BERT | LEGAL-ToBERT |
|---|---|---|---|
| Accuracy | | 0.656 | **0.785** |
| MCC | | 0.559 | **0.727** |
| F1 | Macro | 0.472 | **0.574** |
| | Micro | 0.656 | **0.785** |
| P | Macro | 0.532 | **0.623** |
| | Micro | 0.656 | **0.785** |
| R | Macro | 0.457 | **0.564** |
| | Micro | 0.656 | **0.785** |

**Table 5**
Test results for BUILD dataset.

We also analyzed the performance of our method on each rhetorical role separately. Table 6 shows the precision, recall, and macro F1 score for each rhetorical role. In terms of macro F1 score, LEGAL-ToBERT outperforms LEGAL-BERT in almost each rhetorical role, apart from sentences asserting the petitioner arguments (ARGP), for which, surprisingly, LEGAL-BERT performs 12% better. The two models perform equally well on sentences about not relied precedents (PRENR), sentences presenting the the issue of the debate (ISSUE) and statute sentences (STA). The improvement achieved by LEGAL-ToBERT for all other rhetorical roles ranges from 4% (RPC - ruling sentences by the present court) to 40% (ARGR - sentences asserting the respondent argument).

| RR | LEGAL-BERT | | | LEGAL-ToBERT | | |
|---|---|---|---|---|---|---|
| | F1 | P | R | F1 | P | R |
| PRE | 0.837 | 0.875 | 0.810 | **0.972** | **0.965** | **0.980** |
| FAC | 0.789 | 0.771 | 0.816 | **0.873** | **0.877** | **0.869** |
| RLC | 0.633 | 0.703 | 0.602 | **0.712** | **0.794** | **0.668** |
| ISSUE | **0.890** | **0.926** | 0.859 | 0.886 | 0.876 | **0.898** |
| ARGP | **0.646** | **0.650** | **0.642** | 0.575 | 0.637 | 0.554 |
| ARGR | 0.497 | 0.493 | 0.500 | **0.698** | **0.681** | **0.719** |
| ANA | 0.744 | 0.740 | 0.761 | **0.844** | **0.836** | **0.862** |
| STA | 0.802 | **0.802** | 0.802 | **0.805** | 0.769 | **0.854** |
| PRER | 0.697 | **0.832** | 0.645 | **0.732** | 0.784 | **0.697** |
| PRENR | **0.499** | **0.498** | **0.500** | **0.499** | **0.498** | **0.500** |
| RAT | 0.520 | 0.655 | 0.514 | **0.609** | **0.784** | **0.570** |
| RPC | 0.898 | 0.879 | 0.919 | **0.936** | **0.950** | **0.922** |
| NONE | 0.892 | 0.912 | 0.869 | **0.951** | **0.972** | **0.933** |

**Table 6**
Test results for each rhetorical role of BUILD dataset.

## 5.3. Discussion

Our experiments show that approaches to legal RRC based on LEGAL-ToBERT greatly improve the baselines set by vanilla stand-alone LEGAL-BERT models, in two different languages and legal contexts.

The huge improvement in performances is imputable to the capability of ToBERT models to deal effectively with long documents, by considering and leveraging the relationships between the different sentences of the same legal judgement. Other than this, the relative positional encoding strategy that we applied in the upper layer of our hierarchical transformer allows our approach to take into account the correlation between the rhetorical roles of the individual sentences and their relative position in the document, which provides further hints for the correct classification of a sentence, leveraging legal documents repetitive rhetorical structure.

LEGAL-ToBERT results are particularly surprising in the case of ITA-RhetRoles dataset. This is most reasonably due to the higher amount of data this is composed of, which allows a complex model like ToBERT to reach and exploit its maximum potential, and to the ease of this task, given the repetitiveness of the structure of the documents used. The results achieved on the BUILD dataset are much worse in absolute terms, due to the greater difficulty of the task (much more labels, much less data), but the relative improvement introduced by ToBERT on the baseline is comparable, if not even better, with respect to that achieved on ITA-RhetRoles (MCC improves by 30% in the case of BUILD and by 21% in that of ITA-RhetRoles).

Such promising results invite to employ this model architecture to automate RRC in related applications, giving high hopes of achieving relevant outcomes in many different legal document analysis tasks.

## 5.4. Limitations

While ToBERT models have shown impressive performance on legal RRC benchmarks, we want to highlight some of their main limitations.

**ToBERT models are computationally expensive**. ToBERT models rely on a huge number of parameters, which makes training and fine-tuning much more computationally expensive than other competitive approaches, including CRFs and stand-alone BERT models. This can become a serious limit in terms of scalability and practicality of use in certain applications. For instance, dealing with very long documents (e.g., thousands of sentences) or with documents with very long sentences (e.g., many houndreds of tokens) could become unfeasible without very powerful computational resources, both in terms of time and space complexity.

**ToBERT models require high availability of annotated data**. When running experiments on very small datasets (less than 100 documents), we did not find any advantage in using ToBERT compared to vanilla BERT. These and other experimental results suggest that the effectiveness of automated legal RRC using supervised NLP models is highly affected by the size and complexity

of the dataset and the quality of the annotations. The need for such approaches to have big and high-quality datasets is very restricting, as the availability of such datasets in the legal context is particularly limited for privacy and discretionality reasons.

**ToBERT models do not generalize well to documents longer than those seen during training**. For architectural reasons, ToBERT models are unable to manage effectively documents longer than those seen during training.

**ToBERT models may lack interpretability**. A hierarchical use of transformer-based models introduces a further layer of complexity which makes it even more challenging to interpret model decisions, leading to difficoulties in identify and diagnose errors or biases in model predictions.

**LEGAL-ToBERT models suffer from limited multilingual support**. LEGAL-ToBERT models rely on pretrained language-specific LEGAL-BERT models, which makes it difficoult to apply this approach to multilingual or cross-language tasks. Deploying such models is not easy as it requires the fine-tuning of a BERT model using a huge amount of legal documents in the considered language. Still, our hope is that the availability of legal domain-specific pre-trained models will quickly improve with time, breaking new grounds in many different languages.

## 6. Conclusion and Future Work

In this work we introduced LEGAL-TransformerOverBERT (LEGAL-ToBERT), a novel approach to legal rhetorical roles classification that leverages the power of Hierarchical Transformers and legal-domain-specific BERT models. We also proposed a novel embedding strategy for the top layer encoder of LEGAL-ToBERT, based on the sinusoidal encoding of the document sentences using their relative position in the document instead of the absolute one. Our results provide evidence that this approach allows for a robust and effective framework able to classify efficiently the rhetorical roles of the sentences of long legal documents by taking into account the relationships between them.

We tested the effectiveness of LEGAL-ToBERT on two different datasets. The first one is ITA-RhetRoles, a novel yet confidential dataset, consisting of thousands of documents from the Italian Civil Court Corpus; the second one is the BUILD benchmark dataset, composed of a couple of hundred documents from a various set of Indian courts. This allowed us to diversify our experiments in terms of both language and topic. We showed that LEGAL-ToBERT significantly outperforms vanilla standalone LEGAL-BERT models, on both ITA-RhetRoles and BUILD datasets, improving the macro F1 score by 12% and 22% and the MCC by 21% and 30% respectively.

Future research should aim to extend and improve the approach proposed to other domains and languages. It is also important to address the problem of building robust frameworks in absence of large dataserts, which is most often the case when dealing with the legal domain. On the other hand, we hope that the constant progress in legal NLP will incentivize the collection and the release of increasingly large datasets. Finally, our models are publicly available and ready-to-use, and we ourselves plan to leverage them to enable and improve many downstream applications such as summarization and argument mining of legal documents.

## Acknowledgments

This work is part of Italian nationwide "Giustizia Agile" (Agile Justice) project[4], funded by the Italian Ministry of Justice.

## References

[1] A. Farzindar, G. Lapalme, Letsum, an automatic legal text summarizing system, JURIX (2004) 11–18.

[2] I. Nejadgholi, R. Bougueng, S. Witherspoon, A semi-supervised training method for semantic search of legal facts in canadian immigration cases., in: JURIX, 2017, pp. 125–134.

[3] J. Savelka, K. D. Ashley, Segmenting us court decisions into functional and issue specific parts., in: JURIX, 2018, pp. 111–120.

[4] B. Hachey, C. Grover, Extractive summarisation of legal texts, Artificial Intelligence and Law 14 (2006) 305–345.

[5] P. Kalamkar, A. Tiwari, A. Agarwal, S. Karn, S. Gupta, V. Raghavan, A. Modi, Corpus for automatic structuring of legal documents, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 4420–4429. URL: https://aclanthology.org/2022.lrec-1.470.

[6] V. R. Walker, K. Pillaipakkamnatt, A. M. Davidson, M. Linares, D. J. Pesce, Automatic classification of rhetorical roles for sentences: Comparing rule-based scripts with machine learning., ASAIL@ ICAIL 2385 (2019).

[7] S. R. Ahmad, D. Harris, I. Sahibzada, Understanding legal documents: classification of rhetorical role of sentences using deep learning and natural language processing, in: 2020 IEEE 14th International Con-

[4]More information is available at https://www.unitus.it/it/unitus/mappatura-della-ricerca/articolo/giustizia-agile.

ference on Semantic Computing (ICSC), IEEE, 2020, pp. 464–467.

[8] S. Ghosh, A. Wyner, Identification of rhetorical roles of sentences in indian legal judgments, in: Legal Knowledge and Information Systems: JURIX 2019: The Thirty-second Annual Conference, volume 322, IOS Press, 2019, p. 3.

[9] D. Licari, G. Comandè, ITALIAN-LEGAL-BERT: A Pre-trained Transformer Language Model for Italian Law, in: D. Symeonidou, R. Yu, D. Ceolin, M. Poveda-Villalón, D. Audrito, L. D. Caro, F. Grasso, R. Nai, E. Sulis, F. J. Ekaputra, O. Kutz, N. Troquard (Eds.), Companion Proceedings of the 23rd International Conference on Knowledge Engineering and Knowledge Management, volume 3256 of *CEUR Workshop Proceedings*, CEUR, Bozen-Bolzano, Italy, 2022. URL: https://ceur-ws.org/Vol-3256/#km4law3, iSSN: 1613-0073.

[10] P. Henderson, M. S. Krass, L. Zheng, N. Guha, C. D. Manning, D. Jurafsky, D. E. Ho, Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset, 2022. arXiv:2207.00220.

[11] A. Chriqui, I. Yahav, I. Bar-Siman-Tov, Legal hebert: A bert-based nlp model for hebrew legal, judicial and legislative texts, SSRN preprint:4147127 (2022).

[12] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutsopoulos, LEGAL-BERT: The muppets straight out of law school, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 2898–2904. URL: https://aclanthology.org/2020.findings-emnlp.261. doi:10.18653/v1/2020.findings-emnlp.261.

[13] B. Hachey, C. Grover, A rhetorical status classifier for legal text summarisation, in: Text Summarization Branches Out, 2004, pp. 35–42.

[14] M.-F. Moens, E. Boiy, R. M. Palau, C. Reed, Automatic detection of arguments in legal texts, in: Proceedings of the 11th International Conference on Artificial Intelligence and Law, ICAIL '07, Association for Computing Machinery, New York, NY, USA, 2007, p. 225–230. URL: https://doi.org/10.1145/1276318.1276362. doi:10.1145/1276318.1276362.

[15] M. Saravanan, B. Ravindran, Identification of rhetorical roles for segmentation and summarization of a legal judgment, Artificial Intelligence and Law 18 (2010) 45–76.

[16] V. Malik, R. Sanjay, S. K. Guha, A. Hazarika, S. Nigam, A. Bhattacharya, A. Modi, Semantic segmentation of legal documents via rhetorical roles, 2022. arXiv:2112.01836.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[18] P. Bhattacharya, S. Paul, K. Ghosh, S. Ghosh, A. Wyner, Deeprhole: deep learning for rhetorical role labeling of sentences in legal case documents, Artificial Intelligence and Law (2021) 1–38.

[19] J. Lu, M. Henchion, I. Bacher, B. M. Namee, A sentence-level hierarchical bert model for document classification with limited labelled data, in: Discovery Science: 24th International Conference, DS 2021, Halifax, NS, Canada, October 11–13, 2021, Proceedings 24, Springer, 2021, pp. 231–241.

[20] I. Chalkidis, X. Dai, M. Fergadiotis, P. Malakasiotis, D. Elliott, An exploration of hierarchical attention transformers for efficient long document classification, arXiv preprint arXiv:2210.05529 (2022).

[21] R. Pappagari, P. Zelasko, J. Villalba, Y. Carmiel, N. Dehak, Hierarchical transformers for long document classification, in: 2019 IEEE automatic speech recognition and understanding workshop (ASRU), IEEE, 2019, pp. 838–844.

[22] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[23] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[24] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.

[25] J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, in: J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, volume 24, Curran Associates, Inc., 2011. URL: https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.