TüReuth Legal at SemEval-2023 Task 6: Modelling Local and Global Structure of Judgements for Rhetorical Role Prediction

Henrik Manegold University of Bayreuth Henrik.Manegold@uni-bayreuth.de

Abstract

This paper describes our system for SemEval-2023 Task 6: LegalEval: Understanding Legal Texts. We only participate in Sub-Task (A), Predicting Rhetorical Roles. Our final submission achieves 73.35 test set F1 score, ranking 17th of 27 participants. The proposed method combines global and local models of label distributions and transitions between labels. Through our analyses, we show that especially modelling the temporal distribution of labels contributes positively to performance.

1 Introduction

This paper describes our system for SemEval-2023 Task 6: LegalEval: Understanding Legal Texts. While the shared task hosts 3 sub-tasks, we only participate in sub-task (A), predicting Rhetorical Roles (RRs) from pre-segmented judgments. Refer to the overview paper (Modi et al., 2023) for a detailed description of the task and data.

Briefly, the task is as follows: Given a judgment that is pre-segmented into paragraphs (often containing only 1 sentence) and an inventory of Rhetorical Roles, predict the appropriate RR for every paragraph. RRs structure the judgment into semantically coherent units and provide information on what is discussed in each unit. For example, the RR PREAMBLE denotes formal introductory statements of judgments. In total, there are 13 RRs in this task.

The shared task provides 245 segmented and labelled Indian judgements (in English) for training, 30 segmented and labelled judgments for development, and 50 segmented but unlabelled test set judgments for evaluation. Test set labels are not released by the organisers.

The main challenge of this task is how to integrate document level information and paragraph level information. Since solving the task requires precise understanding of paragraph content Leander Girrbach University of Tübingen leander.girrbach@uni-tuebingen.de

and the context, fine-tuning pre-trained large language models (LMs) is the common starting point. However, LMs can not process long documents such as judgments on standard hardware at once. Therefore, we develop and evaluate a method to use document-level information in order to refine paragraph-level predictions from a fine-tuned LMs. In particular, we model the positions in a document where RRs usually occur, structural constraints that indicate whether some RRs may or may not be predicted when other RRs have already been predicted earlier, and lexical constraints motivated by domain knowledge.

Throughout this paper, we assume the following notation: R is the set of Rhetorical Roles, $r \in R$ denotes some RR, a judgement J consists of a sequence of paragraphs s_1, \ldots, s_T where T is the number of paragraphs in J. Paragraphs are indexed by t (we think of them as a time-series).

2 Related Work

Closest to our method is previous work by the shared task organisers (Malik et al., 2021; Kalamkar et al., 2022). Both works propose to first encode each sentence of a judgement by LM. Then, another sequence processor, such as LSTM or CRF, is applied to predict RRs from the sequence of sentence encodings. Kalamkar et al. (2022) focus on how to arrive at sentence encodings from token encodings computed by a BERT model. On the other hand, Malik et al. (2021) focus on label shift. They train additional models to predict whether 2 consecutive sentences have the same RR. Then, they use this information as feature in their sequence labelling models.

In this work, we continue the approach of Malik et al. (2021) and include more structural modelling. However, in our experiments we found models to predict whether consecutive sentences share the same RR to yield insufficient performance on the negative class (when RRs are different). Most likely, the reason is that, as noted by Malik et al. (2021), most consecutive sentences share the same RR, resulting in very imbalanced data. Therefore, we choose a binary model of whether certain transitions are possible or not instead, which is less expressive but trivial to learn.

Furthermore, treating whole judgments as datapoints for supervised sequence labelling yields only 245 train datapoints for this shared task, which may be problematic because of overfitting. Also, endto-end training of sentence encoders and sequence labelling models is very expensive. Therefore, we do not use any sequence level neural models, but combine scores from individual models during inference.

3 Method

Our method combines sentence-level information with document-level information in order to improve performance of models only using limited contexts. Concretely, we

- Fine-tune a pre-trained language model to predict RRs from individual sentences
- Learn a conditional distribution of RRs given all RRs predicted so far
- Learn a binary score for all ordered pairs of RRs that indicates whether the respective transition from the previous RR to the following RR is allowed
- Learn a continuous distribution over positions in the document that indicates how likely it is for the respective RR to appear at this position in the document
- Hard code lexical rules based on domain knowledge

In the following, we describe each of the individual models in detail. Then, we describe how to combine all these models to assign labels to all paragraphs of a given judgement.

3.1 Models

This section contains descriptions of the different models of local or global document information. Sentence-level LMs, Local Transition Scores, and Domain Knowledge model information about individual or adjacent paragraphs. Position Scores and Global Transition Scores model global distributions or constraints regarding where RRs may appear in a given judgement. **Language Model Scores** We use NLTK (Bird et al., 2009) to sentence tokenize each judgment. Each sentence is assigned the rhetorical role of its corresponding paragraph in the training data. Then, we fine tune the roberta-base model (Liu et al., 2019) for 2 epochs with learning rate $2 \cdot 10^{-5}$ and no weight decay. Given a sentence *s*, the LM computes a probability distribution over RRs.

For inference, in case a paragraph contains multiple sentences, we multiply probabilities for each sentence to get probabilities for the whole paragraph. We refer to the prediction probabilities of RR $r \in R$ for paragraph s_t as $LM(s_t, r)$. LM may also represent an ensemble of LMs by averaging prediction probabilities.

Global Transition Scores We learn conditional Bernoulli distributions $B_{r_{next},A(t)}$ that indicate the probability of a RR given the complete history of RRs predicted so far. The history of previously predicted RRs is represented by a vector $A(t) = [A^k(t) | k \in R] \in \{0,1\}^{|R|}$ of binary variables $A^k(t) \in \{0,1\}$ that indicate whether RR $k \in R$ has been predicted as label for any paragraph s_1, \ldots, s_t .

Intuitively, this distribution models interactions of predicted RRs that block or reinforce prediction of some RRs later in the judgement. For example, after having predicted PREAMBLE and any other RR, it is not possible to predict PREAMBLE again.

Because we cannot empirically estimate parameters for all distributions $B_{r_{next},A(t)}$ due to data sparseness, we estimate parameters of the distributions in the following way: We construct a dataset $X_{\text{global}}, Y_{\text{global}}$ where X_{global} contains all vectors A(t) actually encountered in the dataset and Y_{global} contains binary vectors whose components indicate whether the respective RR appears as label of any paragraph s_{t+1}, \ldots, s_T . Then, we learn a MLP to map binary vectors A(t) to probabilities that indicate how likely it is for any rhetorical role to still appear in the judgement. The MLP is trained by minimising the binary cross entropy loss when predicting labels $\in Y_{\text{global}}$ from X_{global} . This allows us to generalise to vectors A(t) not encountered in the training set.

Local Transition Scores For each ordered pair of RRs $(r_{\text{prev}}, r_{\text{next}}) \in R \times R$ we store whether the respective transition is encountered in the dataset. To align this information with the other probability based scores, we represent transitions that appear in the dataset as a Bernoulli distribution $B_{r_{prev},r_{next}}$ with parameter 1, and we represent transitions that do not appear in the dataset as a Bernoulli distribution $B_{r_{prev},r_{next}}$ with parameter 0.

Furthermore, we learn a Multinomial prior distribution Prior over all RRs that indicates how likely the respective RR is to be the label of the first paragraph s_0 in any judgement.

Position Scores For each RR $r \in R$, we learn a continuous distribution Pos_r over the interval [0; 1]. $\text{Pos}_r(t)$ for $t \in [0; 1]$ indicates the likelihood of RR r appearing at relative position t in the judgment. In reality, there is a finite number of paragraphs $s_1, \ldots, s_t, \ldots, s_T$ in each judgment, meaning relative positions $\frac{t}{T}$ for all $t = 1, \ldots, T$ are discrete and not continuous. To generalise to judgments of different lengths, however, we relax this property and assume continuous relative positions.

Each distribution Pos_r is modelled as a mixture of two Beta distributions, which are defined on the interval [0; 1]. Therefore, each distribution Pos_r has 5 parameters, α_1 and β_1 (the parameters of the first Beta distribution), α_2 and β_2 (the parameters of the second Beta distribution), and $w \in [0; 1]$ (the mixture weight). Then, the density of Pos_r is expressed as

$$\operatorname{Pos}_{r}(t) = \begin{vmatrix} w & \cdot \operatorname{Beta}(t, \alpha_{1}, \beta_{1}) \\ +(1-w) & \cdot \operatorname{Beta}(t, \alpha_{2}, \beta_{2}) \end{vmatrix}$$

We learn parameters of distributions Pos_r for each RR by collecting all relative positions where r appears in any judgement of the dataset. If ris the RR assigned to paragraph s_t in a judgment with T paragraphs in total, the collected value is $\frac{t}{T}$. Then, we map relative positions to one of 100 discrete bins. From these binned relative positions, we compute the empirical CDF (i.e. cumulative histogram) and fit the mixture of Beta distributions by black-box optimisation using the minimize function of the scikit-learn library (Pedregosa et al., 2011) with L-BFGS-B optimization method. The minimised loss is the euclidean distance of the CDF of Pos_r (evaluated at bin boundaries) and the empirical CDF of the RR as described above.

In Figure 1, we show the resulting distributions for some RRs. In particular, we want to highlight that different RRs appear in different places in judgments and we exploit this property by explicitly modelling the temporal distribution of RRs in a judgment.



Figure 1: Distributions Pos_r for different RRs

Domain Knowledge Often, statistical methods may be complemented by domain knowledge in order to refine predictions. This is especially the case in venerable and long-standing fields such as law, where experts have a very good understanding of the subject. In the following, we introduce relevant concepts observed in the given data:

While legal documents may be unstructured (Kalamkar et al., 2022), court decisions always follow the same structure. After a preamble, the court must answer the legal questions raised in the case and reach a final decision based on the facts, the relevant statute law, the case law, and the arguments of the lawyers. Therefore, we implement rules that prevent predicting PREAMBLE when any RR other than NONE was already predicted after PREAMBLE. This applies not only to PREAMBLE, but also prediction of RATIO blocks the prediction of PREAMBLE, FAC, RLC, ISSUES, ARGUMENT BY PETITIONER, ARGUMENT BY RESPONDENT, ANALYSIS, STA, PRECEDENT RELIED and PRECE-DENT NOT RELIED, whereas the prediction of RPC also prevents the prediction of RATIO.

Additionally, we prevent predicting certain RRs when certain conditions are met. First, when one of the strings "JUDGMENT", "J U D G M E N T", "ORDER", "ORDERS", and "O R D E R" appear in a paragraph labelled as PREAMBLE, the next paragraph cannot be labelled as PREAMBLE.

Furthermore, we specify phrases that, if present, enforce or prevent prediction of certain RRs. For example, the phrases "appeal allowed" and "appeal dismissed" enforce prediction of RR RPC, whereas the phrase "learned counsel" prevents the prediction of any RR of PREAMBLE, FACTS, RLC, IS-SUES, STA, PRECEDENT RELIED and PRECEDENT NOT RELIED. The full list of phrases is in Appendix B.

3.2 Inference

When assigning RRs to paragraphs of an unseen judgement at test time, we use all information described in Section 3.1. The judgment consist of pre-segmented paragraphs s_1, \ldots, s_T . We model global information by states S(t, r, A) that consist of the current paragraph index $t \in \{1, \ldots, T\}$, the assigned RR $r \in R$, and the binary vector $A \in \{0,1\}^{|R|}$ that indicates which RRs have already been predicted for paragraphs s_1, \ldots, s_t . The maximum probability path through states is computed by the Viterbi algorithm. In the following, we describe how to compute non-zero state transition scores. Let $S_{\text{prev}} = S(t, r_{\text{prev}}, A)$ and $S_{\text{next}} = S(t+1, r_{\text{next}}, A')$, where A' is identical to A, but the binary indicator for r_{next} is also set to 1 (if not already the case). The transition probability $\Pr[S_{\text{prev}} \rightarrow S_{\text{next}}]$ is calculated as

$$\Pr[S_{\text{prev}} \to S_{\text{next}}] = \begin{bmatrix} \operatorname{LM}(s_t, r_{\text{next}}) \\ \cdot B_{r_{\text{next}}, A} \\ \cdot B_{r_{\text{prev}}, r_{\text{next}}} \\ \cdot \operatorname{Pos}_{r_{\text{next}}}(t+1) \end{bmatrix}$$

In the case of the first paragraph, $B_{r_{prev},r_{next}}$ is replaced with the prior probability $Prior(r_{next})$, r_{prev} is undefined and A contains only 0. To avoid numerical problems, all scores (which are probabilities and therefore ≥ 0) are transformed to log space. Finally, lexical domain knowledge is used to enforce or block certain transitions, i.e. by setting $Pr[S_{prev} \rightarrow S_{next}]$ to 0 irrespective of other scores.

Remember that all transitions not described above have 0 probability. In particular, it is only possible to transition to a state of the next paragraph index. This enforces that each paragraph is assigned exactly one label by each path.

Note that since both T and R are finite, the set of states is also finite. However, it is very large (containing > 10K states for each paragraph index). Therefore, we only consider the 100 states with maximum score for each paragraph index t to make inference efficient. Doing so, we do not notice any performance degradation.

4 **Results**

In this section, we present the final performance of our method and further insights regarding the individual models described in Section 3.1. **Test Set Results** On the test set, our approach achieves 73.35 micro-averaged F1 score.¹ In the shared task ranking, it achieves rank 17 of 27. This score is achieved when using all models described in Section 3.1 and only using train data to fit models. When combining train and development data, we do not get a better result in this case. We assume the reason is the the amount of train data is already sufficient to reach maximum performance of all models except fine-tuning the LMs. Also note that LM scores are averaged prediction probabilities of an ensemble of 5 roberta-base models fine-tuned independently.

Ablation Study In the following, we evaluate the contribution of individual models. Because the number of test set evaluation runs was limited by the organisers and test labels are not released, we report ablations using the development set for evaluation.

First, observe the progression in Table 1. Clearly, the most important part by far is the LM, which extracts the semantic information from sentences. This is a conditio sine qua non. RRs cannot be correctly inferred without knowledge of the text's meaning. Then, Position Scores boost performance by almost 4 points. This shows that modelling global document structure is helpful, especially because Positional Scores can be integrated without much effort into most sequence labelling models. This is not the case for Global Transition Scores, which require keeping track of previously predicted labels and therefore make inference more complicated. Still, Global Transition Scores lead to a performance improvement of almost 1 point. In contrast, Local Transition Scores do not yield further improvement of performance. Most likely, our chosen model, only allowing transitions observed in the training set, is not expressive enough to contribute useful information beyond sentence-level predictions by the LM. Therefore, continuing work by Malik et al. (2021) and including more expressive models of local transitions could be interesting future work. Finally, lexical rules informed by domain knowledge yield another small improvement of a little over 1 point. This is in line with many observations that when given sufficient data, statistical approaches yield better performance, especially because manually coding rules can only reach sufficient recall with a lot of effort. However, the fact that there is improvement shows that hu-

¹We report scores in range 0 to 100 for better readability

LM	Pos	GTS	LTS	DK	RB	LB
\checkmark					72.32	68.01
\checkmark	\checkmark				76.17	75.30
\checkmark	\checkmark	\checkmark			77.01	76.52
\checkmark	\checkmark	\checkmark	\checkmark		77.01	77.04
\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	78.19	77.08

Table 1: Ablation Results (micro-averaged F1 on dev set). Score of submitted model is bold. LM = Language Model, Pos = Position Scores, GTS = Global Transition Scores, LTS = Local Transition Scores, DK = Domain Knowledge. RB = roberta-base, LB = Legal-BERT

mans are still able to identify high-precision rules that are not obeyed by LMs.

In the next analysis step, we want to quantify the contribution of individual models. To this end, we propose the following method: For each individual model M (i.e. LM, Pos, ...) and configuration of used models, we substract the f1 score achieved when not using M from the f1 score achieved when using M. Then, for each individual model M, we calculate the mean over all differences. This score gives an estimate of the performance improvement when adding the respective model to any subset of other models. However, we only evaluate configurations that use LM scores, because not using them does not make sense. The results are:

M	Pos	GTS	LTS	DK
Δ	2.98	2.06	0.25	0.64

where Δ is the average difference between configurations using the model and configurations not using the model.

The results show that all models contribute positively, i.e. no model reduces performance on average. The strongest effect is achieved by the global models. Local models do not yield such improvements, most likely because of the weaknesses discussed above. Also note that the cumulative difference scores very closely match the performance difference between only using LM and using all models in Table 1. Scores for all configurations are in Appendix A.

Choice of LM We experimented with other LMs than roberta-base, which we eventually use for analyses in this paper and our submission to the shared task. Other LMs we tried include but are not limited to T5 (Raffel et al., 2020), SciBERT (Beltagy et al., 2019), and Legal-BERT (Zheng

et al., 2021). Even if using other LMs eventually yields better performance on the test set, the negative results on the development set show that the gains are not consistent. Therefore, to make more use of the capabilities of pre-trained LMs, a different setup (or possibly fine-tuning hyperparameters) than explored here would be necessary.

However, we still want to demonstrate that the results of our analysis are not an artifact of the language model we use, which arguably is the component that influences performance the most. Therefore, we replicate the ablation study from Table 1 for Legal-BERT and also report the results in Table 1.

Here, we can see that the performance of the LM alone is weaker, but in contrast the performance gain resulting from including Position Scores is much higher. Perhaps not surprisingly, this shows that global modelling is more beneficial when using a very strong local model, such as LM, is not possible or feasible. On the other hand, effects of all other models are reduced:

M	Pos	GTS	LTS	DK
Δ	7.15	1.41	0.81	0.17

This shows that the general observations regarding different models agree also for different LMs.

5 Conclusion

In this work, we propose and evaluate different models of global and local judgment structure. Besides using pre-trained language models, the greatest performance gains result from modelling the temporal distribution of labels, i.e. where in the document a label occurs more often. Furthermore, this information can be easily integrated with most sequence labelling methods. Otherwise, we also show that modelling how the possibility of predicting RRs depend on previously predicted RRs and integrating domain knowledge improve performance to a lesser degree. We do not find relevant improvements from restricting transitions between RRs to such transitions that appear in the training data.

As possible next steps, we think it would be interesting to improve performance of LMs, e.g. by showing them more context, and to develop better models of local transitions between RRs. Finally, we make our code available on GitHub.²

²https://github.com/LGirrbach/ Tuereuth-Legal-at-SemEval-Task-6

Acknowledgements

We thank the organisers of the shared task for their efforts and the reviewers for helpful comments and suggestions.

References

- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciB-ERT: A pretrained language model for scientific text. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3615– 3620, Hong Kong, China. Association for Computational Linguistics.
- Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.".
- Prathamesh Kalamkar, Aman Tiwari, Astha Agarwal, Saurabh Karn, Smita Gupta, Vivek Raghavan, and Ashutosh Modi. 2022. Corpus for automatic structuring of legal documents. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 4420–4429, Marseille, France. European Language Resources Association.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.
- Vijit Malik, Rishabh Sanjay, Shouvik Kumar Guha, Shubham Kumar Nigam, Angshuman Hazarika, Arnab Bhattacharya, and Ashutosh Modi. 2021. Semantic segmentation of legal documents via rhetorical roles. *CoRR*, abs/2112.01836.
- Ashutosh Modi, Prathamesh Kalamkar, Saurabh Karn, Aman Tiwari, Abhinav Joshi, Sai Kiran Tanikella, Shouvik Guha, Sachin Malhan, and Vivek Raghavan. 2023. SemEval-2023 Task 6: LegalEval: Understanding Legal Texts. In *Proceedings of the* 17th International Workshop on Semantic Evaluation (SemEval-2023), Toronto, Canada. Association for Computational Linguistics (ACL).
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

LM	Pos	GTS	LTS	DK	Acc.
\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	78.19
\checkmark		\checkmark	\checkmark	\checkmark	76.24
\checkmark	\checkmark	\checkmark		\checkmark	78.12
\checkmark		\checkmark		\checkmark	76.21
	\checkmark	\checkmark	\checkmark	\checkmark	31.05
		\checkmark	\checkmark	\checkmark	10.42
	\checkmark	\checkmark		\checkmark	32.06
		\checkmark		\checkmark	8.02
\checkmark	\checkmark		\checkmark	\checkmark	76.59
\checkmark			\checkmark	\checkmark	72.77
\checkmark	\checkmark			\checkmark	76.14
\checkmark				\checkmark	72.25
	\checkmark		\checkmark	\checkmark	26.95
			\checkmark	\checkmark	28.62
	\checkmark			\checkmark	26.78
\checkmark	\checkmark	\checkmark	\checkmark		77.01
\checkmark		\checkmark	\checkmark		74.68
\checkmark	\checkmark	\checkmark			77.01
\checkmark		\checkmark			74.71
	\checkmark	\checkmark	\checkmark		28.17
		\checkmark	\checkmark		19.56
	\checkmark	\checkmark			28.17
		\checkmark			6.91
\checkmark	\checkmark		\checkmark		76.62
\checkmark			\checkmark		72.84
\checkmark	\checkmark				76.17
\checkmark					72.32
	\checkmark		\checkmark		25.08
			\checkmark		35.26
	\checkmark				25.08

Table 2: Full Ablation Results

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. 2021. When does pretraining help?: assessing self-supervised learning for law and the casehold dataset of 53, 000+ legal holdings. In *ICAIL '21: Eighteenth International Conference for Artificial Intelligence and Law, São Paulo Brazil, June 21 - 25, 2021*, pages 159–168. ACM.

A Full Ablation Results

In Table 2, we list the full ablation results, using an ensemble of 5 independently fine-tuned roberta-base models as LM. Note that we exclude trivial configuration such as not using any model or only using domain knowledge.

B Lexical Rules

In Table 3, we list all phrases that prevent prediction of a certain RR. This means, if a phrase from the list given for a RR appears in a paragraph, the respective RR can not be predicted as RR of the paragraph. All phrases are case sensitive. Furthermore we include the following lexical rules:

- Paragraphs that are all in caps are PREAMBLE or NONE
- Parapgraphs with relative position > 0.9 that inlcude the word "costs" are RPC
- Paragraphs that contain one of "petition" or "appeal" and also contain one of "is dismissed" or "is allowed" are RPC

Rhetorical Role	Phrases
ANALYSIS	"appeal is allowed", "appeal is dismissed", "appeals are allowed", "appeals are dismissed", "ORDER", "appeal allowed", "appeal dismissed", "of the Court was delivered by", "learned counsel for the", "Whether on the facts", "in the case of", "for the offence punishable under section", "held that",
ARG_PETITIONER	"learned counsel for the respondent", "learned counsel for the defendant", "Court held", "in my opinion", "in our opinion", "appeal is allowed", "appeal is dismissed", "appeals are allowed", "appeals are dismissed", "I hold", "We hold", "ORDER", "appeal allowed", "appeal dismissed", "of the Court was delivered by", "Whether on the forter"
ARG_RESPONDENT	"accused pleaded", "learned counsel for the petitioner", "learned counsel for the apellant", "learned counsel for the accused", "Court held", "in my opinion", "in our opinion", "appeal is allowed", "appeal is dismissed", "appeals are allowed", "appeals are dismissed", "I hold", "We hold", "ORDER", "appeal allowed", "appeal
FAC	dismissed', "of the Court was delivered by', "Whether on the facts', "Government Pleader", "learned counsel", "learned counsel for the petitioner", "learned counsel for the apellant", "Court held", "in my opinion", "in our opin- ion", "appeal is allowed", "appeal is dismissed", "appeals are allowed", "appeals are dismissed", "I hold", "We hold", "ORDER", "learned senior counsel", "prosecution failed", "appeal allowed", "appeal dismissed", "argues", "of the Court was delivered be".
ISSUE	by, "Government Pleader", "accused pleaded", "learned counsel", "learned counsel for the petitioner", "submit", "submis", "Court held", "in my opinion", "in our opinion", "appeal is allowed", "appeal is dismissed", "appeals are allowed", "appeals are dismissed", "I hold", "We hold", "however", "commissioner", "fine", "ORDER", "learned senior counsel", "prosecution failed", "appeal allowed", "appeal dismissed",
NONE	"argues", "of the Court was delivered by", "accused pleaded", "Court held", "appeal is allowed", "appeal is dismissed", "ap- peals are allowed", "appeals are dismissed", "I hold", "We hold", "commissioner", "ORDER", "prosecution failed", "appeal allowed", "appeal dismissed", "Whether on the facts".
PREAMBLE	"accused pleaded", "learned counsel", "learned counsel for the petitioner", "learned counsel for the apellant", "in my opinion", "in our opinion", "appeal is allowed", "appeal is dismissed", "appeals are allowed", "appeals are dismissed", "I hold", "We hold", "learned senior counsel", "prosecution failed", "appeal allowed", "appeal dismissed", "argues", "of the Court was delivered by", "from the judgment", "Whether on the facts"
PRE_NOT_RELIED	"Government Pleader", "learned counsel", "learned counsel for the petitioner", "learned counsel for the apellant", "appeal is allowed", "appeal is dismissed", "ap- peals are allowed", "appeals are dismissed", "commissioner", "ORDER", "learned senior counsel", "prosecution failed", "appeal allowed", "appeal dismissed", "ar- gues", "of the Court was delivered by", "Whether on the facts",
PRE_RELIED	"Government Pleader", "accused pleaded", "learned counsel", "learned counsel for the petitioner", "learned counsel for the apellant", "appeal is allowed", "appeal is dismissed", "appeals are allowed", "appeals are dismissed", "commissioner", "ORDER", "learned senior counsel", "prosecution failed", "appeal allowed", "appeal dismissed", "argues", "of the Court was delivered by", "Whether on the facts",
RATIO	"Government Pleader", "accused pleaded", "learned counsel", "learned counsel for the petitioner", "learned counsel for the apellant", "Court held", "appeal is allowed", "appeal is dismissed", "appeals are allowed", "appeals are dismissed", "ORDER", "learned senior counsel", "appeal allowed", "appeal dismissed", "argues", "of the Court was delivered by", "Whether on the facts",
RLC	"Government Pleader", "accused pleaded", "learned counsel", "learned counsel for the petitioner", "learned counsel for the apellant", "submit", "submis", "appeal is allowed", "appeal is dismissed", "appeals are allowed", "appeals are dismissed", "learned senior counsel", "appeal allowed", "appeal dismissed", "argues", "of the Court was delivered by", "Whether on the facts",
RPC	"accused pleaded", "learned counsel", "learned counsel for the petitioner", "learned counsel for the apellant", "learned senior counsel", "argues", "of the Court was delivered by", "Whether on the facts",
STA	"Government Pleader", "accused pleaded", "learned counsel", "learned counsel for the petitioner", "learned counsel for the apellant", "Court held", "appeal is allowed", "appeal is dismissed", "appeals are allowed", "appeals are dismissed", "I hold", "We hold", "ORDER", "learned senior counsel", "prosecution failed", "appeal allowed", "appeal dismissed", "argues", "of the Court was delivered by", "Whether on the facts",

Table 3: List of phrases that prevent predicting a certain RR.