Legal Data Analysis Lab SS2022 Report

Abdullah Al Sefat, Irtiza Chowdhury
TUM Informatics

ABSTRACT

In this paper we attempt to assign rhetorical roles to legal court documents (i.e. sequential sentence classification). We explore various methods of sequential sentence classification as well as sequential span classification where we attempt to label spans of sentences with a label. A Semi-Markov CRF based model remodeled the problem into a sequential span classification paradigm, outperforming the baseline on span level and, for some labels, on a sentence level. For sequential sentence classification Machine Reading Comprehension based method slightly outperformed the Sequence to Sequence based method on the same dataset. The Sequence to Sequence method was applied on multiple datasets and we reached the conclusion that it performed better with larger training data.

1 INTRODUCTION

Rhetorical role of a sentence is the function it serves in a document. The task of assigning labels or roles to sentences based on the semantic function it serves is called rhetorical role labeling. In the context of legal documents, rhetorical role labeling is of paramount importance. It is worth mentioning that, legal case documents tend to be quite long as well as unstructured. However, if legal documents are labeled with appropriate rhetorical roles then multiple downstream tasks can be performed such as semantic search [4] and summarizing [9].

2 BACKGROUND AND RELATED WORK

Several methodologies in rhetorical role labelling were explored during the timeframe of the project to devise a solution that could not only benefit from prior knowledge, but also contribute with future possibilities. Some of the literature research conducted has been detailed below.

2.1 Identification of Rhetorical Roles of Sentences in Indian Legal Judgments [1]

CITE The authors took a deep learning approach to automatically identify rhetorical roles of sentences across five legal domains, using a dataset of 50 documents and 7 rhetorical roles. A Hierarchical Bidirectional LSTM+CRF classifier was used with initialization of sentence embeddings using sent2vec from 53K court case documents. They concluded that Neural models with pretrained embeddings performed significantly better. Performance across domains was consistent to the inter-annotator agreement, showing the advantage of neural models in requiring no hand-crafting of features to get impressive results.

2.2 DeepRhole: deep learning for rhetorical role labeling of sentences in legal case documents [2]

A comprehensive analysis of several deep learning models with tweaks in their architectures was presented by the authors. They claim substantial improvement over prior methods and robustness of theirs, including Transformer models based on BERT and Legal-BERT. In general, the baseline models with CRF were outclassed by the neural models. Improvement in scores also came from using Law2Vec pretrained word embeddings than using other pretrained word embeddings, as this initialization made the models more aware of the domain. Using a CRF on a hierarchical BiLSTM model showed minor improvements, explained by the large sequences and few documents in the dataset, so the CRF was unable to capture the transition scores.

2.3 Hierarchical Neural Networks for Sequential Sentence Classification in Medical Scientific Abstracts [5]

The authors emphasized the need to consider context which is generally unseen in traditional sentence classification. They presented a hierarchical sequence labelling network (HSLN) to capture contextual information from surrounding sentences in a sequential sentence classification task. The proposed model consisted of several components which are elaborated in the following sections. Their best model had higher F-scores than the previously published best results for all datasets. The key component from ablation analysis was the context enriching layer which optimized the label sequence. The difficult labels to classify were similar in nature, but overall better performance with room for generalization was noted.

2.4 Sequential Span Classification with Neural Semi-Markov CRFs for Biomedical Abstracts [11]

Sequential sentence classification methods were prone to mislabel longer continuous sentences in abstracts. This issue was handled by the authors by altering the problem statement to sequential span classification where a span would consist of continuous sentences. Neural Semi-Markov CRF (SCRF) was used to consider all possible spans of various lengths. The authors pointed out the critical problem with previous methods such as HMMs, BiLSTM+CRF and SciBERT which was the inconsistency in performance on longer spans as they mainly deal with a small context. The proposed model contained BiLSTM layer to generate the span vectors from the contextualized sentence vectors followed by the SCRF layer to learn the labelling of the sequences of spans. The method achieved best

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

scores for both micro-averaged sentence and span-F scores in both datasets, showing that sequential span classification could boost classification accuracy of spans with larger number of sentences.

3 METHODOLOGY/DESIGN

Several approaches to sequential sentence classification were explored and analyzed for the task of rhetorical role labelling. The following sections highlight the methodology and specifics behind the design.

3.1 Sequential Span Classification

Yamada et al. [11] concluded that the chances of mislabelling longer continuous sentences were much higher in traditional sequential sentence classification. While their work focused on datasets related to scientific abstracts, a similar concept is applicable for legal corpora as long sentences are common characteristics in them. A Hierarchical Sequential Labelling Network (HSLN) [5] was constructed for sequential span classification on legal text documents. The model consists primarily of five components described below and highlighted in Figure 8 of Appendix A.

3.1.1 Word embedding layer. Compared to the prior work done on scientific abstracts, Legal-BERT embeddings were more suitable for our project. Alternatively, a sent2vec model published by Bhattacharya et al. [1] was also utilized to directly get sentence embeddings from the documents, alleviating the need for a word embedding layer.

3.1.2 Sentence Encoding Layer. The sentence was then encoded into a new vector from the embedding vectors generated in the previous layer. A Bi-LSTM was used to process the sequence of embedding vectors resulting in a sequence of hidden states for each sentence. Attention-based pooling was applied to get the final representation of every sentence in the form of a vector.

3.1.3 Context Enriching Layer. From the encoded sentence vectors in a document, the contextual information from surrounding sentences was encoded using a Bi-LSTM layer to create a new sequence of vectors.

3.1.4 Span Extraction Layer. Diverging from the regular architecture, an additional span extraction layer was necessary to get span representations from the contextualized sentence vectors. **Bidirectional span extractors** concatenate two different representations, one forward and one backward, of the starting and ending sentences in a span. Yamada et al. [11] extends this by also concatenating hidden states of the previous as well as next sentence. **Self-attentive extractor** computes span representations by generating an unnormalized attention score for each word in the document. Spans representations are computed with respect to these scores by normalising the attention scores for words inside the span. **Endpoint span extractors** simply combine embeddings of the sentences as endpoints. Multiple combinations can be used as representations, including elementwise addition, multiplication or division of embeddings.

3.1.5 Label Sequence Optimization Layer. Conditional Random Fields can capture certain implicit patterns in sequence of sentence labels. While their *emission* scores represent the label given a sentence, *transition* scores signify the probability of a label given the previous label in the sequence of sentences, capturing their dependencies. During training, the probability of the gold label sequence is maximized. In the testing phase, given an input sequence, the corresponding sequence of predicted labels is chosen as the one that maximizes the score, computed via the Viterbi algorithm. [5]

Neural Semi-Markov CRFs operate similarly, but now outputs segment $\boldsymbol{s} = \{s_1, s_2, ..., sp\}$ of an input sequence $\boldsymbol{X}$. They learn parameters to maximize the log-likelihood of the correctly labelled sequence of spans given the sequence of sentences. Each segmented sequence of sentences is assigned a label to create spans $\boldsymbol{s_j} = t_j, u_j, y_j$ where t_j = start position u_j = end position and y_j = label. As shown in Figure 1, given five sentences, Semi-Markov CRFs consider spans of all possible lengths to produce the correctly (in red) labelled sequence [11].

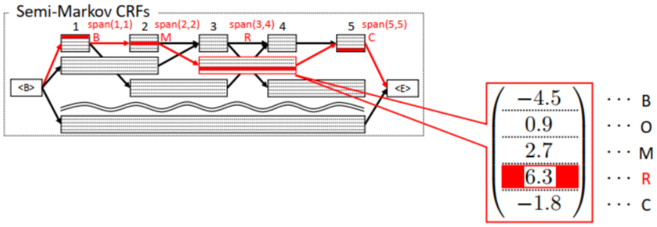


Figure 1: Correctly labelled sequence of spans using Semi-Markov CRFs

3.2 Alternative Approaches to Sequential Sentence Classification

In this section we will briefly discuss about the further methods of sequential sentence classification that we ventured and their prospect for sequential span classification.

3.2.1 Machine Reading Comprehension. The typical Machine Reading Comprehension(MRC) task is to make machines read text and answer questions about the corresponding text [12]. MRC tasks require data to consist of a triplet of Question, Context and Answer. The sequential sentence classification task can be modeled into an MRC task. In figure 2 the data preparation of MRC is shown. We have a set of documents $D = d_1, d_2, ..., d_m$ where each document d_i has an arbitrary number of sentences. Let $Q = q_1, q_2, ..., q_l$ be a set of questions where l is the number of labels. The questions are labels and their definitions concatenated. The concatenation of labels are done for providing context. Then for each question and each document we have an answer which indicates the sentences in a document which belong to a certain class of label. In figure 3 an example of MRC question answer data is shown. For a question q_1 and document d_1 with sentences $x_1, .., x_n$ the sentences denoted in the color green have gold labels corresponding to the label of q_1 and the corresponding answer a_{11} has 1s for those sentences and 0s otherwise.

For MRC sequential sentence classification the backbone of the model remains almost identical and therefore we will not go too in depth. At first the MRC question answering dataset is generated. The question and each sentence is concatenated. The concatenated sentence is fed to LegalBERT [3] for tokenization and to obtain

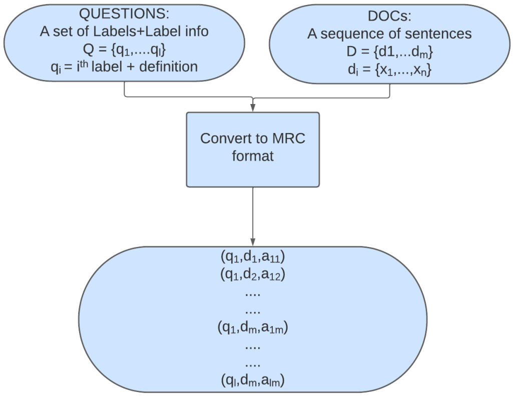


Figure 2: Data preparation of MRC task

q_1= {Label_1 + definition}	d_1 = {x_1,x_2,x_3,x_4,x_5,....x_p,x_{p+1},...x_n}	a_{11} = {0,1,1,1,0,.....,1,1,...,0}

Figure 3: Example of an MRC question answer data for sentence level classification

token level representation. The token embeddings of each sentence are then sent to a bi-LSTM and further goes through attention based pooling to obtain a sentence level encoding. The sentence encodings are sent to a bi-LSTM for context enrichment. This is followed by a fully connected layer.

3.2.2 Sequence to Sequence. Sequence to sequence learning using neural networks was first introduced by Google researchers where they performed machine translation on sequences of English text to obtain sequences of French text [10]. The sequence to sequence learning paradigm follows an encoder-decoder structure. The input sequence is fed into the encoder which is typically an RNN(LSTM, bi-LSTM or GRU) in order to get a latent representation. The latent representation is then fed into the decoder to sequentially obtain the desired sequence. The output of each timestep at the decoder is fed as input to the next timestep. Various sequence tagging tasks can be transformed to a sequence to sequence task [8]. This is what motivated us to transform the sequential sentence classification task into a sequence to sequence task. For our task at hand a sequence of input sentence is fed to the encoder and the decoder predicts a sequence of labels for the sequence of input sentences as shown in the figure 4.

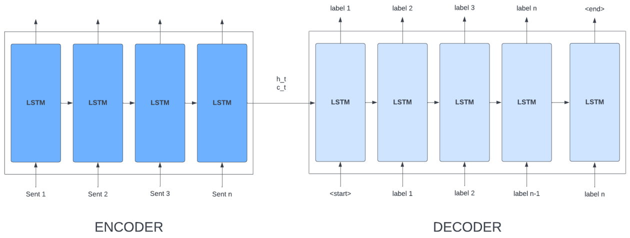


Figure 4: A schematic view of sequential sentence classification using a seq2seq model

We have performed sequential sentence classification using the Text to Text Transfer Transformer(T5) model which is a pretrained sequence to sequence model [7]. For converting a sequence tagging task into a seq2seq task the target of the sequence can be of various formats such as tags+input, tags only and sentinel+tags [8]. In this paper we have used tags only as targets where the tags are the labels of the sentences.

4 DATASETS & EVALUATION

4.1 Legal Corpora

For our experiments we have used two different legal corpora. The first dataset was curated by Bhattacharya et. al. [1] where the authors used legal judgements from the Supreme Court of India, sampling 50 documents from 53,210 documents according to the top 5 domains proportionally. The documents were then annotated to have 7 different rhetorical roles.

The second dataset was curated by Kalamkar et. al [6] which was also based on Indian legal text documents, contained 265 case documents from Supreme Court, High Courts and District Courts. Thirteen different rhetorical labels were present across over 26,304 sentences. Additionally, PubMed 20k RCT was used to test the validity of the methods.

4.2 Baseline Experiments

In order to effectively compare the effect of our proposed methodologies and draw conclusive results, some baselines had to be established for the models using the available datasets. A Hierarchical Bi-LSTM [1] baseline obtained **0.857** accuracy, **0.797** macro F1 and **0.857** weighted average F1 on the Bhattacharya et. al. dataset. For the Kalamkar corpus, a HSLN+CRF [5] achieved **0.765** accuracy, **0.497** macro F1 and **0.761** weighted average F1. Legal-BERT produced better results for both models and datasets. However, sentence embeddings from the referred sent2vec model showed improvement for the Kalamkar dataset, with a weighted sentence F1 score of **0.791** compared to **0.761** from the baseline. The details of the evaluation metrics are given in the appendix.

4.3 Sequential Span Classification Results

An additional metric, Span F1 [11], was applied as regular sentence level F1 did not fully capture performance on a span level. For example, five sentences with gold label sequence, *Facts-Analysis-Analysis-Issue-None* and a prediction, *Facts-Analysis-Issue-Issue-None* would have sentence-F1 of **0.8**. However, as only two spans out of four were correctly identified, Span-F1 would only be **0.5**. It can be defined as a harmonic mean of precision and recall based on a perfect match of span-by-span labels. Experiments on the corpora are summarized below.

4.3.1 Kalamkar et al. After testing out the three span extraction methods, the Endpoint extractor with four combinations of embeddings produced better results. The proposed **HSLN+Semi-Markov CRF** method outperformed the baseline (HSLN) model in both key evaluation metrics of Span-F1 and weighted Sentence-F1, **0.380** and **0.790** respectively, compared to Span-F1 **0.280** and sentence F1 **0.761** from the baseline. While the Span-F1 showed significant improvement, even resulting sentence F1 was comparable to the currently published best score by the authors. In Figure 5, the performance on a span level can be compared for different labels in the validation set. Legal-BERT was used for the embeddings, and a key hyperparameter was the span length which indicates the maximum number of sentences in a span. From our analysis, a higher span length generally worsened the performance of the model not only in terms of metrics, but also resources requirements. Therefore, the best results achieved came from a span length of 2, but the argument could be made that longer training time and more resources could allow for better performance with longer span lengths. Figure 6 of Appendix A shows that our best model accomplished better results for certain labels compared to the baseline even at a sentence level, while model of longer span lengths could barely keep up. Additional results are noted in Figure 7 of Appendix A

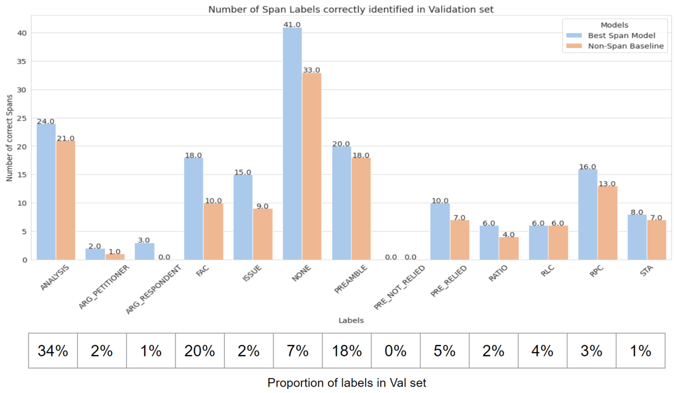


Figure 5: Span level performance of best proposed model vs baseline model

4.3.2 Bhattacharya et al. A significant improvement was not evident for this dataset. The best performing model resulted in a Span-F1 of **0.120** compared to **0.114**, barely outperforming and sometimes even underperforming for some labels in the corpus. As pointed out earlier, this corpus contained only one-fifth the number of documents than the other which could contribute to its comparatively weaker performance. Even the authors noted that a simpler Hierarchical Bi-LSTM+CRF model had been more suitable than a complex one during their experiments [2].

4.4 Experimental Results of MRC and Seq2Seq

We conducted the MRC based experiment only on the dataset of Kalamkar et al. We were able to train the model for only 5 epochs before reaching resource usage limit. We obtained **0.497** weighted average F1 and **0.265** macro average F1 score which is much lower than the baseline model's performance on the Kalamkar et al. dataset.

We trained the T5 Seq2Seq model on both the datasets. The T5 model obtained a **0.363** accuracy, **0.246** macro F1 and **0.356** weighted average F1 on the Bhattacharya et. al. dataset after training for 6 epochs. For the Kalamkar dataset the T5 model obtained an accuracy of **0.503** macro F1 of **0.219** and weighted F1 of**0.448** after training for 5 epochs. We can see that on the Kalamkar dataset the MRC apporoach slightly outperforms the Seq2Seq approach. However the difference is very low and both the models were not trained for a reasonable number of epochs to make a strong claim for the MRC approach. The scope of hyperparameter tuning was out of question since the models could not be trained for enough epochs as they had very high computational requirements with limited provisioning of compute resources.

The T5 model can be seen to perform better on the Kalamkar et al. dataset over the Bhattacharya et al. dataset. The reason behind this is most likely due to the fact that there are simply more training samples in the Kalamkar dataset. The Bhattacharya dataset has rather diminutive with only 50 documents. Additionally we used the PubMed20k RCT dataset to perform sequential sentence classification using the T5 model where we got **0.752** accuracy, **0.682** macro F1 and **0.747** weighted average F1. The PubMed20K RCT has a very large number of documents and the results are much better. This corroborates the fact that the Bhattacharya dataset performed very poorly due to having very low number of training data compared to the other datasets.

5 CONCLUSION

While sequential span classification shows promise in the legal domain, the complexity and size of the corpus complicate the process. Semi-Markov CRFs should be able to consider spans of all possible lengths, but the sheer number of sentences compared to abstracts does not allow that. Compared to an average number of 13 sentences per document in PubMed 20k corpus, the number of sentences in our chosen legal corpora was around 150. As the number of possible spans is $nC2 + n$, span space explodes in our context. Higher span lengths could not be properly experimented with because it became more resource intensive, reaching only half the number of epochs within usage limits for longer lengths. Word or sentence level augmentation can be used to solve issues related to label imbalance and small corpus.

Provided better resources, further hyperparameter tuning can be conducted. Optimal span length can be tested from the average length of spans in the corpus. Alternatively, an order invariant approach to get the same results regardless of the order of the input components can be explored to make the current model non-autoregressive. Algorithmic changes to Semi-Markov CRFs can also be a potential experiment.

Limited resource provisioning in the Google Colab platform certainly a big drawback. The MRC based method was very resource intensive as the dataset increases by l times in size where l being the number of labels. Each epoch took between 2.5 to 3.5 hours limiting use to train the model up to maximum 5 epochs. This restricted us from realizing the learning potential of the model let alone hyperparameter tuning. For the Seq2Seq model the maximum length of the input sequence was a constraint. Since we used a pretrained T5 model we were not able provide entire document or a chunk of the document as an input to label span of sentences.

If resource constraints can be solved the models can be further trained to reasonable number of epochs. In the future the MRC based method for detecting spans of sentences can be implemented. A custom hierarchical Seq2Seq model can be implemented for working with documents to label sequential span of sentences.

REFERENCES

[1] Paheli Bhattacharya, Shounak Paul, Kripabandhu Ghosh, Saptarshi Ghosh, and Adam Wyner. 2019. Identification of Rhetorical Roles of Sentences in Indian Legal Judgments. (2019). https://doi.org/10.48550/ARXIV.1911.05405

[2] Paheli Bhattacharya, Shounak Paul, Kripabandhu Ghosh, Saptarshi Ghosh, and Adam Wyner. 2021. DeepRhole: deep learning for rhetorical role labeling of sentences in legal case documents. *Artificial Intelligence and Law* (2021), 1–38.

[3] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. *CoRR* abs/2010.02559 (2020).

[4] Racchit Jain, Abhishek Agarwal, and Yashvardhan Sharma. 2020. Spectre@AILA-FIRE2020: Supervised Rhetorical Role Labeling for Legal Judgments using Transformers. In *FIRE*.

[5] Di Jin and Peter Szolovits. 2018. Hierarchical neural networks for sequential sentence classification in medical scientific abstracts. *arXiv preprint arXiv:1808.06161* (2018).

[6] Prathamesh Kalamkar, Aman Tiwari, Astha Agarwal, Saurabh Karn, Smita Gupta, Vivek Raghavan, and Ashutosh Modi. 2022. Corpus for Automatic Structuring of Legal Documents. (2022). https://doi.org/10.48550/ARXIV.2201.13125

[7] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. (2019). https://doi.org/10.48550/ARXIV.1910.10683

[8] Karthik Raman, Iftekhar Naim, Jiecao Chen, Kazuma Hashimoto, Kiran Yalasangi, and Krishna Srinivasan. 2022. Transforming Sequence Tagging Into A Seq2Seq Task. (2022). https://doi.org/10.48550/ARXIV.2203.08378

[9] M. Saravanan, B. Ravindran, and S. Raman. 2008. Automatic Identification of Rhetorical Roles using Conditional Random Fields for Legal Document Summarization. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*. https://aclanthology.org/I08-1063

[10] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. *CoRR* abs/1409.3215 (2014).

[11] Kosuke Yamada, Tsutomu Hirao, Ryohei Sasano, Koichi Takeda, and Masaaki Nagata. 2020. Sequential span classification with neural semi-Markov CRFs for biomedical abstracts. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 871–877.

[12] Changchang Zeng, Shaobo Li, Qin Li, Jie Hu, and Jianjun Hu. 2020. A Survey on Machine Reading Comprehension: Tasks, Evaluation Metrics and Benchmark Datasets. (2020). https://doi.org/10.48550/ARXIV.2006.11880

A APPENDIX

A.1 Performance and Evaluation Metrics

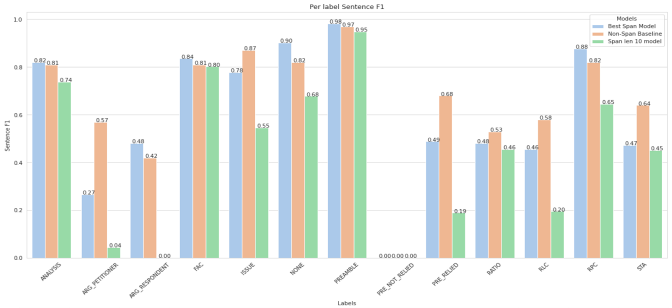
$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \tag{1}$$

$$Macro\ F1 = \frac{\Sigma Per\ Class\ F1}{Total\ Number\ of\ Classes} \tag{2}$$

$$Weighted\ Average\ F1 = \Sigma(Per\ Class\ F1\ *\ Per\ Class\ Weight) \tag{3}$$

Where, Per Class F1 = True Positive / (True Positive + 0.5(False Positive + False Negative)) and Per class weight is the proportion of a class' support value relative to the sum of all support values.

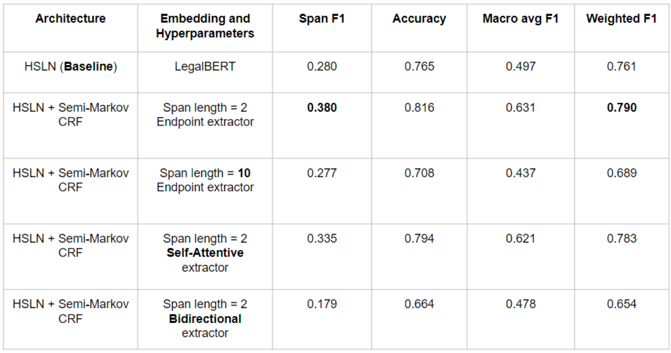
A.2 Figures for Sequential Span Classification



34%	2%	1%	20%	2%	7%	18%	0%	5%	2%	4%	3%	1%

Proportion of labels in Val set

Figure 6: Sentence-F1 comparison of best Semi-Markov CRF model, same model with span length of 10 and baseline



Architecture	Embedding and Hyperparameters	Span F1	Accuracy	Macro avg F1	Weighted F1
HSLN (**Baseline**)	LegalBERT	0.280	0.765	0.497	0.761
HSLN + Semi-Markov CRF	Span length = 2 Endpoint extractor	**0.380**	0.816	0.631	**0.790**
HSLN + Semi-Markov CRF	Span length = **10** Endpoint extractor	0.277	0.708	0.437	0.689
HSLN + Semi-Markov CRF	Span length = 2 **Self-Attentive** extractor	0.335	0.794	0.621	0.783
HSLN + Semi-Markov CRF	Span length = 2 **Bidirectional** extractor	0.179	0.664	0.478	0.654

Figure 7: Experimental results on Kalamkar corpus

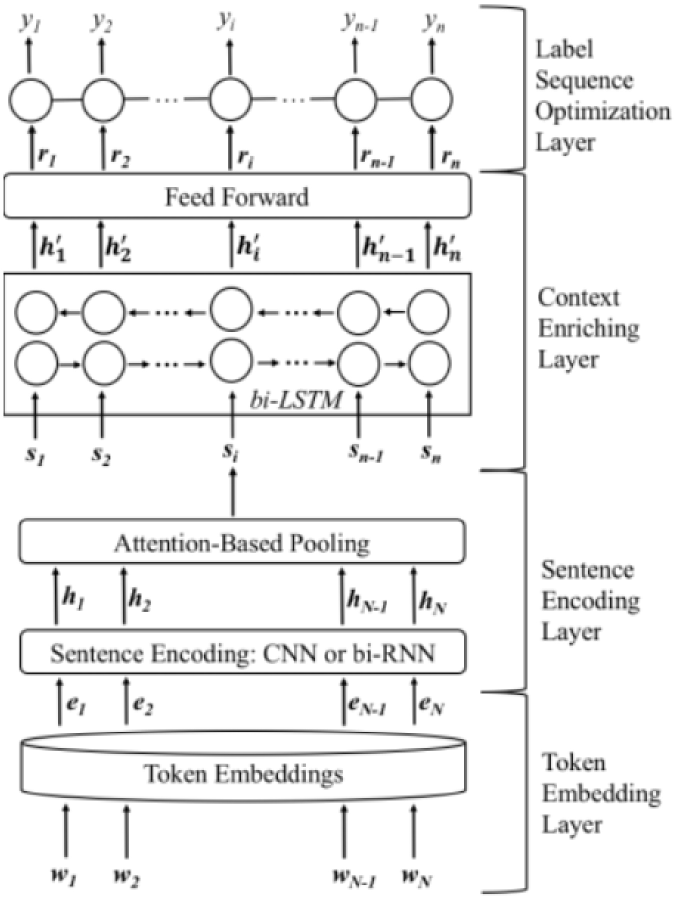


Figure 8: Model architecture of HSLN by Jin et al. [5]