June 2021

Accounting for Sentence Position and Legal Domain Sentence Embedding in Learning to Classify Case Sentences

Huihui Xu [a,1], Jaromir Savelka [b,2] and Kevin D. Ashley [a,c,d,3]
[a] *Intelligent Systems Program, University of Pittsburgh*
[b] *School of Computer Science, Carnegie Mellon University*
[c] *Learning Research and Development Center, University of Pittsburgh*
[d] *School of Law, University of Pittsburgh*

Abstract. In this paper, we treat sentence annotation as a classification task. We employ sequence-to-sequence models to take sentence position information into account in identifying case law sentences as issues, conclusions, or reasons. We also compare the legal domain specific sentence embedding with other general purpose sentence embeddings to gauge the effect of legal domain knowledge, captured during pre-training, on text classification. We deployed the models on both summaries and full-text decisions. We found that the sentence position information is especially useful for full-text sentence classification. We also verified that legal domain specific sentence embeddings perform better, and that meta-sentence embedding can further enhance performance when sentence position information is included.

Keywords. Information retrieval; Natural language processing; Annotation; Embedding

1. Introduction

As an initial step toward automatically generating comprehensible legal summaries, we have been exploring machine learning (ML) methods for classifying sentences of legal cases in terms of issues a court addresses, its conclusions of those issues, and its reasons for so concluding (IRCs). In previous work, we have experimented with different models—both traditional machine learning and deep learning—to identify these types of sentences in both summaries and full texts. While we demonstrated that those models can identify IRC types of sentences to some extent, the task remains challenging for machine encoding.

In this paper, we employ supervised ML based on a larger annotated dataset, 1049 pairs of full text cases and summaries in which sentences have been manually annotated in terms of IRCs. We also explore if two new techniques, sentence embeddings pretrained on large quantities of legal texts and taking account of sentence order, help machine annotation of legal cases.

[1] huihui.xu@pitt.edu
[2] jsavelka@cs.cmu.edu
[3] ashley@pitt.edu

In attempting to leverage the power of state-of-the-art sentence embeddings, pre-trained on legal texts, we hypothesize that the broader contextual information associated with the sentence embeddings will improve performance.

We also hypothesize that taking sentence ordering information into account will improve the classifier's performance. In regular meetings with our two third-year law student annotators to resolve differences concerning annotations, we noticed that they tended to rely on ordering information to mark up certain types of sentences. For example, annotators would look for conclusions following issues or at the end of a case. We wondered if ML could also employ such position information.

1.1. Extracting Issues, Reasons, and Conclusions

The ultimate goal of our work is to enable an intelligent system to help end users assess a case's potential relevance by effectively and efficiently conveying some important substantive information about the case. Human-prepared legal summaries are available through various on-line legal service providers. For example, the CanLII Connects website[4] of the non-profit Canadian Legal Information Institute,[5] features summaries of legal decisions prepared by members of Canadian legal societies.

Based on the experience of CanLII Connects, summaries as short as three sentences could be even more effective in a legal IR interface. This raises a practical question: "What can a three-sentence case summary provide?". Legal argument triples, IRCs, may be the answer. Issues, reasons, and conclusions form the skeleton of case briefs, a legal writing technique for summarizing cases that has long been taught in American law schools. Thus, the potential utility of summarizing cases in terms of issues, conclusions, and reasons seems clear.

Based on our annotation experience, the human-prepared CanLII summaries regularly include issues raised by the courts, the conclusions reached, and reasons connecting them. Those summaries also include some procedural information, descriptions of facts, statements of legal rules, case citations and explanations, and other information. Since the expert legal summarizers act as an intelligent and well-informed filter on importance, it made sense to leverage their expertise by annotating their summaries rather than the full texts. CanLII has provided 28,733 paired cases and human-prepared summaries for purposes of this research. The cases cover a variety of kinds of legal claims and issues presented before Canadian courts.

1.2. Hypotheses

We try to answer two long-standing questions in the Artificial intelligence and Law field: first, whether legal language is so unique that the legal pre-trained models would assist downstream legal natural language processing tasks and which tasks; second, whether sentence position information helps a model as it appears to help human annotators.

We investigate how well the classification models perform based on sentence embeddings, and annotate full texts of cases and summaries in terms of issues, reasons, and conclusions. As noted, we examine two hypotheses in this paper:

[4] `https://canliiconnects.org/en`
[5] `https://www.canlii.org/en/`

(1) A model would perform better when incorporating sentence position information.
(2) A model would perform better when incorporating specific legal domain knowledge.

2. Related Work

2.1. Word and Sentence Embeddings

Word embeddings, dense vector representations trained with neural language models, capture some linguistic relationships between words and assist with various natural language processing tasks. See, e.g., [1]. Researchers further explored word meta-embeddings for operations such as concatenation, SVD, and 1toN [2]. Experiments in [3] proved that averaging different sources of word embeddings has similar effects as concatenating those embeddings. Researchers in [4] used three types of autoencoders to learn meta-embeddings of words.

Similarly, sentence embedding is the dense vector representation of a sentence. Sentence embedding provides information about larger contexts of words. [5] introduced Sentence-BERT in 2019; they used siamese and triplet network structures to derive fixed-sized 768 dimensional vector representations for input sentences. Google Research developed the Universal Sentence Encoder in 2018 [6]. The encoder has two model architectures: one based on transformer architecture and the other on Deep Averaging Networks (DAN). Both transfer input sentences into fixed 512 dimensional sentence embeddings. Both Sentence-BERT and Universal Sentence Encoder are state-of-the-art sentence embeddings.

In the legal domain, words may have different semantic meanings than in other domains. For example, 'sentence' means the judgment that a court formally pronounces after finding a criminal defendant guilty.[6] In order to address this, we employed Legal-BERT, a BERT model trained on legal domain sentences [7]. Legal-BERT was pre-trained on the entire Harvard Law case corpus from 1965 to present, comprising 3,446,187 legal decisions across all federal and state courts [7]. [7]

2.2. Argument Mining and Summarization

Extracting propositions, premises, conclusions, and nested argument structures [8,9] is an active research topic in the legal argument mining field. Rhetorical and other roles that sentences play in legal arguments have been employed for legal argument mining [10]. Citing information and fact patterns [11,12] that effect the strength of a side's claim in special legal domains are also being explored. Segmenting legal text by functions [13,14], and by topic [15] or by linguistic analysis [16,17,18] are some initial steps for dissecting a legal document.

Researchers have applied legal argument mining to the task of summarizing legal cases. In [19], the authors propose an unsupervised algorithm that incorporates legal domain knowledge, such as rhetorical roles sentences play in a legal document. [20] have summarized Japanese judgments in terms of issues, conclusions, and framings. Our legal

[6] `https://www.law.cornell.edu/wex/sentence`
[7] The pre-trained Legal-BERT model can be found here: `https://huggingface.co/zlucia/legalbert`

argument triples have a similar structure but are more understandable types than those tailored to Indian or Japanese legal judgements. In addition, in our work a set of case summaries prepared by legal experts is used to extract argument triples from the full case texts.

3. Dataset

Our type system for labeling sentences in legal cases comprises:

1. **Issue** – Legal question which a court addressed in the case.
2. **Conclusion** – Court's decision for the corresponding issue.
3. **Reason** – Sentences that elaborate on why the court reached the Conclusion.

We treat all non-annotated sentences as non-IRC sentences.

Two hired third-year law school students annotated sentences from the human-prepared summaries to identify and annotate the issues, reasons, and conclusions. Both students have annotated 1049 randomly selected pairs from the 28,733 case/summary pairs available. The total number of sentences from the corresponding full texts is 215,080, which is significantly more than the corresponding summaries' 11,496 sentences.

Both annotators followed an 8-page detailed Annotation Guide prepared by the third author, a law professor, in order to mark-up instances of IRC sentence types in both the summaries and full texts of cases. The annotators worked on successive batches of summaries using the Gloss annotation environment developed by the second author. After annotating each batch, the annotators resolved any annotation differences in regular Zoom meetings attended by the first and third authors.

The procedure for annotating the full texts of cases differs from annotating the summaries. The Annotation Guide instructs annotators to search the full text of the case for those sentences that are most similar to the annotated summary sentences and to assign them the same labels (i.e., Issue, Conclusion, or Reason) as in the summaries. Annotators may pick terms or phrases from the annotated summary sentences as anchors to search for corresponding sentences in the full texts. Annotators do not need to read the full text of the case if they find the corresponding sentences. The Guide warns that there may not be an exact correspondence between the annotated sentences in the summary and those in the full text of the case. This is fairly common, because human summarizers tend to edit selected sentences in the full case texts. For example, a human summarizer may combine some shorter sentences into a longer one.

By using the summaries' annotations as anchors to target corresponding sentences in the full text, we attempted to leverage the summarizers' work in selecting important sentences and the annotators' work in marking up some of those full texts sentences as issues, conclusions, or reasons. We developed this strategy to expedite the full text annotation process, since it would be much more time-consuming and costly if annotators had to read the full texts of cases. The strategy is based on the observation that sentences of summaries stem from those in the full texts. The strategy also helps us to confirm the mapping relationship between summaries and full texts, which is a step towards generating summaries automatically.

Cohen's κ [21] is used to measure the degree of agreement between two annotators after their independent annotations. The mean of Cohen's κ coefficients across all

Figure 1. Distribution of annotated IRC type sentences in 1049 summaries (left) and full texts (right).

Table 1. Descriptive statistics of the resulting dataset. We report the basic descriptive statistics of each type in both summaries and full texts. The lengths of the summary and the full texts are also included in the table.

	Summary			Full text		
	Min.	Max.	Mean	Min.	Max.	Mean
Issue	3 tokens	140 tokens	27.96 tokens	3 tokens	427 tokens	36.80 tokens
Reason	3 tokens	257 tokens	26.61 tokens	3 tokens	229 tokens	31.36 tokens
Conclusion	2 tokens	289 tokens	20.40 tokens	2 tokens	314 tokens	28.55 tokens
Length of text	1 sents	90 sents	10.96 sents	9 sents	2411 sents	205.03 sents

types for summaries is 0.734, and the mean for full texts is 0.602. According to [22], both scores indicate substantial agreement between annotators about the sentence type. For the summary annotation, the mean of Reason agreement is the lowest among those three types. Annotating Reasons is more challenging since they are entwined with case facts. The agreement scores of full texts are lower than the summaries' scores, since sentences from summaries and full texts are not in a one-to-one mapping. This increases the difficulty of full text annotation.

Figure 1 reports the distributions of final consensus labels from summaries and full texts. The most frequent label is the non-IRC label for both summaries and full texts. The second most frequent label is the Reason label for both summaries and full texts. The label distribution is aligned with our observation: Reasons tend to be more elaborated than Issues and Conclusions.

The descriptive statistics of the processed dataset are shown in Table 1. The average number of sentences in a full text is 205.03, while the range of the full text length is quite large. Comparatively, the average number of sentences in summaries is 10.96 which, as expected, is much shorter than full texts. We also observe that the average length of Issues is the highest in both summaries and full texts.

4. Experiment

4.1. Models

We use Sentence-BERT [5], Universal Sentence Encoder(USE) [6], and Legal-BERT [7] to encode sentences from summaries and full texts into a semantic space. Each sentence

then becomes a fixed sized vector. Each document is comprised of a series of converted sentence vectors.

Sentence-BERT uses two BERT models with tied weights and adds a pooling operation to the output to derive fixed sized sentence embedding. We chose the 'all-mpnet-base-v2' model trained on a dataset of over 1 billion pairs.[8] This model encodes sentences into 768-dimensional vectors and has achieved competitive performance over different datasets. USE takes a tokenized string and outputs a fixed 512-dimensional vector as sentence embedding.[9] Legal-BERT was trained on the entire Harvard Law case corpus. In order to derive the fixed sized sentence embedding, we simply keep the output of the last pooling layer of this model as the sentence embedding. The dimension of the Legal-BERT sentence embedding is also 768.

The Long Short-Term (LSTM) neural network [23], a variant of a recurrent neural network (RNN), can deal with arbitrary lengths of input. A traditional RNN does not perform well on long sequences due to the problem of vanishing gradients. LSTM tackles the problem by incorporating different gates. Bidirectional LSTM consists of two separate LSTMs: one takes an input from right to left; the other one from left to right.

We also examined the effect of one of the meta-sentence embedding techniques. Averaging is one of the commonly used meta-embedding techniques. It simply requires averaging different sources of embeddings. According to [24], averaging has similar performance to concatenation while taking less time and resources in terms of meta-word embedding. We extend this idea to the sentence embedding. We construct two types of meta-sentence embeddings: Legal-BERT + USE and Legal-BERT + Sentence-BERT. Both types of meta-sentence embeddings are 768-dimensional.

4.2. Experimental Design

This work employs two designs which differ as to the way in which the sentence embeddings are fed into the bidirectional LSTM model: 1) Single time step is associated with a sentence, and no sentence position information is provided. 2) Fixed sized document matrices are input into the model; each time step is associated with a sentence, where sentence position information is provided. We refer to [14] for the padding procedure. For example, since the maximum length of full texts is 2411, we transferred each full text document into a 2411×768 matrix when using Sentence-BERT embedding. The shorter case will be padded to the maximum length. In this paper, we chose pre-padding over post-padding since [25] demonstrates that pre-padding for LSTM performs substantially better than post-padding. Figure 2 shows the structure of the models and the main difference between the two designs. Our rolled LSTM reads one sentence at each time step. The returning arrow (left) represents multiple time steps for "with position information". "Without position information" (right) involves only a single time step.

We split the dataset into training, validation, and test sets. The training set comprises 70% of 1049 cases; the validation and test sets each have 15% of the cases. The data is fed into the bidirectional LSTM model with 256 units and a dropout rate of 0.2. Categorical cross-entropy loss function and Adam optimizer are used for optimizing the model. The initial learning rate is set to $1e^{-3}$ and reduced at factor 0.1 if the validation loss has stopped decreasing with a patience of 20. The training procedure will be stopped when

[8] `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`

[9] `https://tfhub.dev/google/universal-sentence-encoder/4`

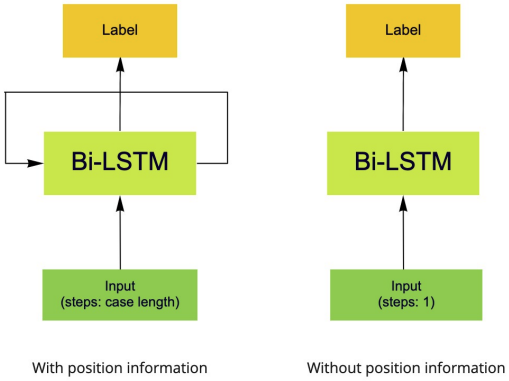

Figure 2. Main difference of two experiment designs: with and without sentence position information.

the validation accuracy has not been increased in 20 epochs. Validation accuracy is used to select the best model.

5. Results and Discussion

The results of the two experimental designs are shown in Table 2. The first 5 rows of the table show how well the model performs on the summary and the full text without position information; the next 5 rows show the performance of the model when incorporating sentence position information. All the numbers are reported as F_1 scores.

5.1. Without Position Information vs. With Position Information

For summaries, the model performs better on identifying Reasons with position information no matter which sentence embedding is used. However, Legal-BERT and Legal-BERT + USE sentence embeddings with position information do not have better performance in terms of Issue and Conclusion classification. The average F_1 scores of all sentence embeddings are higher when the model digests sentence position information at the same time.

For the full text sentence classification, the pattern is much clearer: Issue and Reason can be more easily identified by the model when including sentence position information. Sentence-BERT and Legal-BERT embeddings do not perform better with position information in terms of Conclusion classification. The average F_1 scores are better when the model is fed with sentence position information.

5.2. Domain Specific Sentence Embedding vs. General Purpose Sentence Embedding

Legal-BERT sentence embedding achieves the best performance on Issue, Reason and Conclusion on both summary and full text, if the model was not fed sentence position information. Legal-BERT sentence embedding performs better than other types of sentence embeddings when the model takes sentence position information into account except on classifying Issues in summaries.

Sentence-BERT sentence embedding is the second best embedding on most of the classification tasks, while USE is the second best on full text Reason classification.

Table 2. Results of classification on summaries and full texts with and without position information. All the results are reported as F_1 scores.

	Summary (without position information)					Full text (without position information)				
	Issue	Reason	Conclusion	Non-IRC	Ave.	Issue	Reason	Conclusion	Non-IRC	Ave.
SBERT	0.69	0.62	0.68	0.62	0.66	0.17	0.04	0.44	0.97	0.40
Legal-BERT	0.74	0.68	0.73	0.67	0.70	0.30	0.13	0.49	0.98	0.47
USE	0.55	0.58	0.61	0.61	0.59	0.15	0.06	0.36	0.97	0.39
Legal-BERT+USE	0.73	0.68	0.74	0.68	0.71	0.25	0.14	0.48	0.98	0.46
Legal-BERT+SBERT	0.71	0.70	0.69	0.67	0.69	0.29	0.12	0.47	0.98	0.46
	Summary (with position information)					Full text (with position information)				
	Issue	Reason	Conclusion	Non-IRC	Ave.	Issue	Reason	Conclusion	Non-IRC	Ave.
SBERT	0.73	0.69	0.69	0.65	0.69	0.30	0.06	0.40	0.97	0.43
Legal-BERT	0.69	0.75	0.75	0.66	0.71	0.36	0.14	0.47	0.98	0.49
USE	0.67	0.65	0.64	0.60	0.64	0.31	0.08	0.36	0.97	0.43
Legal-BERT+USE	0.71	0.72	0.72	0.67	0.71	0.41	0.18	0.49	0.98	0.51
Legal-BERT+SBERT	0.76	0.72	0.72	0.70	0.72	0.38	0.20	0.49	0.98	0.51

5.3. Meta-Sentence Embedding vs. Singular Sentence Embedding

For summaries, Legal-BERT + USE and Legal-BERT + SBERT improve model performance on Issue identification with position information. Those two sentence embeddings have tied or better performance on Reasons without position information.

For full texts, meta-sentence embedding substantially improves the performance on Issue, Reason and Conclusion when position information is included. Without position information, the meta-sentence embedding does not show improved performance.

Generally speaking, meta-sentence embeddings coupled with position information show higher increases in performance on full texts than on summaries.

5.4. Error Analysis and Limitations

We present a brief error analysis for comparing the errors between including position information and not including position information when using Legal-BERT sentence embedding. In the test set, the model has $F_1 = 0.30$ on Issues for the full texts without position information as opposed to $F_1 = 0.36$ when including position information. We read some examples that both experimental methods get right, and some instances that only the model fed with position information correctly classified. We noticed that without position information the model tends to select only Issue sentences with certain sentence structure, like "This is an appeal...". With position information, the model would be able to pick up Issue sentences relying less on sentence structure. For example, "The charge arises out of... " has an implicit semantic cue regarding the type of the sentence. The position information provides additional information to help the model to make the correct classification.

We verified the importance of position information in terms of the model classification performance on full texts. However, this pattern does not apply to some types of sentences in summaries, like Issue sentences. It seems like the position information in

summaries is not as reliable as in full texts. We found that the model ignores some Issue instances that appear in the middle of the summaries.

Compared to our prior work [26], the F_1 scores across all types decrease substantially. In [26], we obtained F_1 scores of 0.58, 0.15, and 0.53 on Issue, Reason and Conclusions, respectively. Our expectation that the performance would improve after training on more data was not confirmed. Several reasons could contribute to this result: first, the initial learning rates are different; this will lead to different performance. Second, noisy data also increase along with the increase of data.

6. Conclusion and future work

We analyzed the effect of sentence position information and legal domain specific sentence embedding in a task of labelling case sentences in terms of legal argument triples. We found that the sentence position information does assist the model to perform better, especially for full texts. We also verified that legal domain specific sentence embedding performed better on this legally intensive task than the other general purpose sentence embeddings. Meta-sentence embedding that inherits benefits from general purpose sentence embedding and legal sentence embedding can outperform its components when the position information is incorporated. The result suggests a promising path to annotate legal documents automatically. This is also a step towards automatically generating succinct legal summaries since the model can identify the important sentences.

This work is subject to certain limitations as well. As mentioned before, paradoxically, the overall performance on full texts tended to decrease with the larger training set. For future work, we will explore a more effective model to improve the performance, such as by introducing additional linguistic features and their semantic values.

Acknowledgement

This work has been supported by grants from the Autonomy through Cyberjustice Technologies Research Partnership at the University of Montreal Cyberjustice Laboratory and the National Science Foundation, grant no. 2040490, FAI: Using AI to Increase Fairness by Improving Access to Justice. The Canadian Legal Information Institute provided the corpus of paired legal cases and summaries. Computation resources are provided by the Center for Research Computing at the University of Pittsburgh.

References

[1] Turian J, Ratinov L, Bengio Y. Word representations: a simple and general method for semi-supervised learning. In: Proc. 48th Ann. Mtg. of the Association for Computational Linguistics; 2010. p. 384-94.

[2] Yin W, Schütze H. Learning word meta-embeddings. In: Proc. 54th Ann. Mtg. of the Association for Computational Linguistics (Volume 1: Long Papers); 2016. p. 1351-60.

[3] Coates J, Bollegala D. Frustratingly Easy Meta-Embedding – Computing Meta-Embeddings by Averaging Source Word Embeddings. In: Proc. 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, V. 2. New Orleans, Louisiana: Association for Computational Linguistics; 2018. p. 194-8. Available from: `https://aclanthology.org/N18-2031`.

[4] Bollegala D, Bao C. Learning word meta-embeddings by autoencoding. In: Proc. 27th Int'l Conf. on Computational Linguistics; 2018. p. 1650-61.

[5] Reimers N, Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:190810084. 2019.

[6] Cer D, Yang Y, Kong Sy, Hua N, Limtiaco N, John RS, et al. Universal sentence encoder. arXiv preprint arXiv:180311175. 2018.

[7] Zheng L, Guha N, Anderson BR, Henderson P, Ho DE. When Does Pretraining Help? Assessing Self-Supervised Learning for Law and the CaseHOLD Dataset. arXiv preprint arXiv:210408671. 2021.

[8] Feng V, Hirst G. Classifying arguments by scheme. In: Proc. 49th Ann. Mtg of the Association for Computational Linguistics: Human language technologies; 2011. p. 987-96.

[9] Mochales R, Moens M. Argumentation mining. Artificial Intelligence and Law. 2011;19(1):1-22.

[10] Saravanan M, Ravindran B. Identification of rhetorical roles for segmentation and summarization of a legal judgment. Artificial Intelligence and Law. 2010;18(1):45-76.

[11] Bansal A, Bu Z, Mishra B, Wang S, Ashley K, Grabmair M. Bex F, editor. Document Ranking with Citation Information and Oversampling Sentence Classification in the LUIMA Framework; 2016.

[12] Falakmasir M, Ashley K. Utilizing Vector Space Models for Identifying Legal Factors from Text. In: JURIX; 2017. p. 183-92.

[13] Savelka J, Ashley K. Segmenting U.S. Court Decisions into Functional and Issue Specific Parts. In: Proc. 31st Int. Conf. on Legal Knowledge and Information Systems, Jurix; 2018. p. 111-20.

[14] Savelka J, Westermann H, Benyekhlef K, Alexander CS, Grant JC, Amariles DR, et al. Lex rosetta: transfer of predictive models across languages, jurisdictions, and legal domains. In: Proc. 18th Int'l Conf. on Artificial Intelligence and Law; 2021. p. 129-38.

[15] Lu Q, Conrad J, Al-Kofahi K, Keenan W. Legal document clustering with built-in topic segmentation. In: Proc. 20th ACM int'l conf. Info. and knowledge management; 2011. p. 383-92.

[16] Grover C, Hachey B, Korycinski C. Summarising legal texts: Sentential tense and argumentative roles. In: Proc. HLT-NAACL 03 Text Summarization Workshop; 2003. p. 33-40.

[17] Farzindar A, Lapalme G. Legal text summarization by exploration of the thematic structure and argumentative roles. In: Text Summarization Branches Out; 2004. p. 27-34.

[18] Wyner A, Mochales-Palau R, Moens M, Milward D. Approaches to text mining arguments from legal cases. In: Semantic processing of legal texts. Springer; 2010. p. 60-79.

[19] Bhattacharya P, Poddar S, Rudra K, Ghosh K, Ghosh S. Incorporating domain knowledge for extractive summarization of legal case documents. In: Proc. 18th Int'l Conf. on Artificial Intelligence and Law; 2021. p. 22-31.

[20] Yamada H, Teufel S, Tokunaga T. Building a corpus of legal argumentation in Japanese judgement documents: towards structure-based summarisation. Art Int and Law. 2019;27(2):141-70.

[21] Cohen J. A coefficient of agreement for nominal scales. Educational and psychological measurement. 1960;20(1):37-46.

[22] Landis J, Koch G. The measurement of observer agreement for categorical data. Biometrics. 1977:159-74.

[23] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997;9(8):1735-80.

[24] Coates J, Bollegala D. Frustratingly Easy Meta-Embedding–Computing Meta-Embeddings by Averaging Source Word Embeddings. arXiv preprint arXiv:180405262. 2018.

[25] Dwarampudi M, Reddy N. Effects of padding on LSTMs and CNNs. arXiv preprint arXiv:190307288. 2019.

[26] Xu H, Savelka J, Ashley KD. Toward summarizing case decisions via extracting argument issues, reasons, and conclusions. In: Proc. 18th Int'l Conf. on Artificial Intelligence and Law; 2021. p. 250-4.