Toward Summarizing Case Decisions via Extracting Argument Issues, Reasons, and Conclusions

Huihui Xu
Intelligent Systems Program
University of Pittsburgh
USA
huihui.xu@pitt.edu

Jaromir Savelka
School of Computer Science
Carnegie Mellon University
USA
jsavelka@andrew.cmu.edu

Kevin D. Ashley
Intelligent Systems Program,
University of Pittsburgh
USA
ashley@pitt.edu

ABSTRACT

In this paper, we assess the use of several deep learning classification algorithms as a step toward automatically preparing succinct summaries of legal decisions. Short case summaries that tease out the decision's argument structure by making explicit its *issues*, *conclusions*, and *reasons* (i.e., argument triples) could make it easier for the lay public and legal professionals to gain an insight into what the case is about. We have obtained a sizeable dataset of expert-crafted case summaries paired with full texts of the decisions issued by various Canadian courts. As the manual annotation of the full texts is prohibitively expensive, we explore various ways of leveraging the existing longer summaries which are much less time-consuming to annotate. We compare the performance of the systems trained on the annotations that are manually ported to the full texts from the summaries to the performance of the same systems trained on annotations that are projected from the summaries automatically. The results show the possibility of pursuing the automatic annotation in the future.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; *Retrieval models and ranking*; Similarity measures; • **Applied computing** → **Law**; Annotation.

KEYWORDS

Information retrieval, argument mining, legal analysis, relevant sentences, summarization

ACM Reference Format:
Huihui Xu, Jaromir Savelka, and Kevin D. Ashley. 2021. Toward Summarizing Case Decisions via Extracting Argument Issues, Reasons, and Conclusions. In *Eighteenth International Conference for Artificial Intelligence and Law (ICAIL'21), June 21–25, 2021, São Paulo, Brazil.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3462757.3466098

1 INTRODUCTION

The ability to automatically prepare succinct summaries of legal decisions could contribute to making legal source materials more accessible to the lay public. This depends, however, on whether the summaries capture the gist of the argument in the decision. In prior work, we proposed that such case summaries could be generated by extracting legal argument triples (IRC triples) including: 1) the major *issues* a court addressed in the case, 2) the court's *conclusion* with respect to each issue, and 3) the court's *reasons* for reaching the conclusion.

In [23], we evaluated whether a machine learning (ML) model can identify the components of legal argument triples in summaries prepared by legal professionals. We applied traditional ML algorithms (random forest variations) and deep neural network models (LSTM, CNN and FastText) to identify the sentence components of IRC triples in legal summaries and to the task of binary classifying sentences (IRC vs. non-IRC) in the summaries and corresponding full text decisions. While the performance on the summaries was promising, the performance on the full texts was quite poor.

In this work, we have substantially increased the size of the annotated data set of full case texts compared to the prior work. We focus on applying deep learning algorithms (LSTM, CNN), including pre-trained transformer models (RoBERTa, CNN-BERT) with different loss functions to deal with the continuing challenge of data imbalance in our training set. We report the results of applying the different kinds of neural models on cases' full texts after training with manually-mapped human-annotated sentences from the summaries and analyze the effects of using different loss functions and embeddings. We also report results of a proof-of-concept experiment that applied automatically mapped human-annotated sentences from the summaries to the full-texts in order to classify argument triples. If this succeeds, we would not need to manually annotate the full texts. It would suffice to manually annotate the summaries, automatically map those summary annotations to the full texts, and train a model directly on the full texts.

2 RELATED WORK

Argument mining research in the legal domain has focused on extracting propositions, premises, conclusions, and nested argument structures [16], argument schemes such as by example [6], rhetorical and other roles that sentences play in legal arguments [1, 19], stereotypical fact patterns that strengthen a side's claim (i.e., legal factors) in domains like trade secret law [4], reasons or warrants in arguments citing facts or principles [21], functional parts of legal decisions such as analysis or conclusions [20], and segments by topic [13] or by linguistic analysis [5, 7, 22].

We aim to identify legal argument triples and employ them to succinctly summarize case summaries. Yamada, et al. [24] have summarized Japanese judgments in terms of issues, conclusions, and

ICAIL'21, June 21–25, 2021, São Paulo, Brazil

ACM ISBN 978-1-4503-8526-8/21/06…$15.00
https://doi.org/10.1145/3462757.3466098

framings. The legal argument triples we seek to employ are simpler types, which have not been tailored to Japanese legal judgements. In addition, we make use of a set of case summaries prepared by human experts to assist with extracting argument triples from the full case texts, a resource not employed in the cited work.

3 DATA SET

We defined the components of legal argument triples as follows:

(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.

All non-annotated sentences are treated as non-IRC sentences.

Two paid third-year law school students annotated sentences from the human-prepared summaries to identify the issues, reasons, and conclusions. Both students annotated 574 randomly selected pairs from the 28,733 case/summary pairs that were available to us. The total number of sentences from the corresponding full texts is 120,707, which is significantly more than the summaries' 7,484 sentences. The total number of sentences from full texts are significantly more than sentences from summaries.

Both annotators followed an 8-page Annotation Guide prepared by the third author, a law professor, in order to mark-up instances of IRC sentence types in the summaries. Using the Gloss annotation environment of the second author, the annotators worked on successive batches of summaries during a series of weeks. After annotating each batch, the annotators resolved any coding 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. For each annotated sentence in the summary, the Annotation Guide instructs annotators to search the full text of the case for those sentences that are most similar to the summary sentence and to assign them the same label (i.e., Issue, Conclusion, or Reason) as in the summary. Annotators may pick terms or phrases from the annotated summary sentences and search for corresponding sentences in the full texts. If the annotators find corresponding sentences, they do not need to read the full text of the case. The Guide warns the annotators 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 makes sense; having selected sentences in the full case texts to include in the summary, the summarizers probably edited them, for example, by combining some short sentences.

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 sentences as issues, conclusions, or reasons. We developed this strategy to expedite the process of full text annotation which would be much more time-consuming and costly if performed directly on the full texts. 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. This in turn helps us to develop the heuristic for automatic mapping.

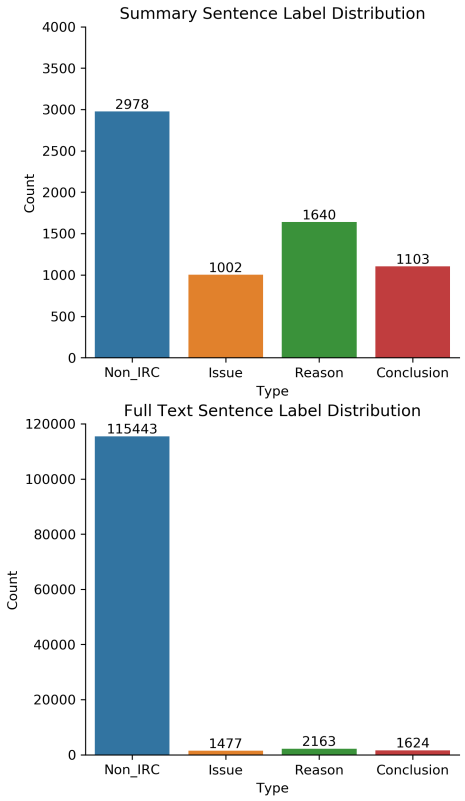

Figure 1: Distribution of annotated IRC type sentences in summaries (top) and in full texts (bottom).

Cohen's κ [2] is used to measure the degree of agreement between two annotators after their independent annotations of each batch of summaries. The mean of Cohen's κ coefficients across all types for summaries is 0.709, and the mean for full texts is 0.827. According to [11], both scores indicate substantial agreement between annotators about the sentence type. For the summary annotation, the mean of Reason agreement is the lowest and Issue and Conclusion's are the highest. Reasons are more challenging since they tend to include descriptions of case facts. The agreement scores of full texts are higher than the summaries' scores. Since the full text annotation took place after reconciling disagreements for the summary annotation, the full text scores were expected to be higher.

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.

4 EXPERIMENTS

We aim to leverage state-of-the-art deep learning models to identify the role a sentence may play in a case as an Issue, Conclusion, or Reason. For our experiments, we used 80% of the full texts as the training set, 10% as the validation set and 10% as the test set.

In this section, we present the details of the models, including convolutional [10] and recurrent neural networks [14], BERT-based neural networks [12] and a hybrid neural model combining a convolutional neural network with BERT embedding [8]. We also experiment with different loss functions for those neural networks, including cross entropy loss, F1 loss and focal loss.

4.1 Model Architectures

Convolutional Neural Networks. Convolutional neural networks (CNN) utilize convolutional filters to extract local features. Originally applied to computer vision tasks, CNNs have also achieved a high level of performance in sentence classification tasks. [9].

In our study, we use the settings of hyperparameters of filters from [9]: filter sizes of 3, 4, and 5 with 100 for each size of filter. In other words, the models are looking for tri-grams, 4-grams and 5-grams in sentences.

Long Short-Term Memory Networks. Long Short-Term Memory (LSTM) networks, a different RNN architecture, overcomes the vanishing gradient problem by employing a cell to control removing or adding information throughout the whole training process [14].

GloVe [17] is an unsupervised learning algorithm for obtaining vector representations for words[1]. We used "glove.6B.100d" as pretrained word embeddings to feed into the LSTM model, where the vectors were trained on 6-billion tokens and have 100 dimensions. Dropout is also adapted to the LSTM model.

BERT-based Neural Networks. Google AI Language introduced Bidirectional Encoder Representations from Transformers (BERT) in 2018 [3]. Instead of using single word embedding like GloVe, BERT takes the context into account by using bidirectional pre-training for language representations. This pre-training method is intended to better grasp contextual meaning of a language than single-directional pre-training.

RoBERTa [12] replicates BERT training using an improved training methodology with more data and computational resources. For our study we used RoBERTa in its default configuration.

Convolutional Neural Network with BERT embedding. CNN with BERT embedding takes BERT-pretrained embeddings as input and feeds them into a CNN model for classification. Unlike GloVe pretrained word embedding, BERT-pretrained embedding is not a static embedding. As sentences are fed in, it produces the word embeddings in real time.

We combine the two models: a BERT-based model and a CNN classification model. The encoded text passes through the BERT model first and produces BERT embeddings. The dimension of the BERT embedding (768) is higher than that of GloVe pre-trained word embedding (100). Other hyperparameters remain the same as for the CNN-only model.

4.2 Automatic mapping

As mentioned in Section 3, the human annotation of full texts utilizes manual mapping: annotators used key words from annotated sentences in original longer summaries to find corresponding sentences in full texts. Even without actually reading the full case, annotators still need to read contextual information around a sentence to confirm the mapping and IRC type.

[1]https://nlp.stanford.edu/projects/glove/

Table 1: Comparison of manually annotated IRC summary sentences with top 1 automatically ranked full-text sentences (Sentence-BERT embedding with cosine similarity)

Issue	
Manual	**Damage** to both vehicles **exceeded** the insurance **deductibles** and both parties **claim damages** against each other for the amount of the **deductibles**.
Rank 1	The **damages** to both the truck and the car **exceeded** the $500.00 insurance **deductible**. [...]
Reason	
Manual	The plaintiff should have taken more **appropriate** measures to avoid the accident
Rank 1	Even if Schmidt concluded that Henry was going to proceed into his path, he had more **appropriate** alternatives than locking his brakes and turning to the right.
Conclusion	
Manual	**Fault** for this **accident** was **attributed 10%** to the defendant and **90%** to the plaintiff.
Rank 1	I **attribute 10%** of the **fault** in this **accident** to Henry and **90%** to Schmidt.

We undertook a proof-of-concept experiment to assess if a strategy of automatic mapping could make the process more efficient in the future. The idea is to employ sentence embedding to map annotated summary sentences to full texts. Sentence embedding can represent a sentence and capture semantic information as vectors. Cosine similarity is used to examine the degree of similarity between sentences in annotated summaries and full texts.

Sentence-BERT Embedding. Sentence embedding techniques represent the entire sentence and semantic information as vectors. Sentence-BERT is a modification of the BERT neural model that uses siamese and triplet networks to produce semantically meaningful sentence embeddings [18]. Sentence-BERT has achieved high levels of performance in measuring the similarity of sentential arguments in [15]. Considering the size of our data set, we chose to use the BERT base model for sentence embeddings with 768 dimensions.

We calculated the cosine similarity score for annotated sentences from a summary to every sentence in its corresponding full text. All the similarity scores are ranked in descending order, and only the top 5 sentences are selected as useful. The remaining sentences are marked as non-IRC type sentences.

There are reasons to believe that the automatically mapped sentences bear useful similarities to manually mapped ones. Table 1 shows examples comparing manually mapped IRC sentences and automatically mapped sentences. The top ranked sentences often include the same key words as the manual sentences do. However, the Reason sentences that the algorithm prefers have fewer overlapping keywords than Issue and Conclusion.

5 RESULTS

Table 2 reports scores for the classification on the test split of the full texts. The left side of the table reports the results of training on the full text sentences corresponding to the manually mapped human-annotated sentences from the summaries.

We tested LSTM, CNN, RoBERTa and CNN-BERT with three different loss functions: cross-entropy loss, focal loss and F1 loss.

Table 2: Scores for test set on both manually mapped full text sentences and automatically mapped full text sentences by using LSTM, CNN, RoBERTa, CNN-BERT. The abbreviations of Issue, Reason and Conclusion are I, R, and C. The suffixes are -P(precision), -R(recall). Ave-F1 stands for the average of class-wise F1 scores.

	Training on manually mapped data										Training on automatically mapped data									
	I-P	I-R	I-F1	R-P	R-R	R-F1	C-P	C-R	C-F1	Ave-F1	I-P	I-R	I-F1	R-P	R-R	R-F1	C-P	C-R	C-F1	Ave-F1
LSTM(cross-entropy)	0.72	0.49	**0.58**	0.35	0.09	0.15	0.72	0.42	**0.53**	0.42	0.19	0.41	0.26	0.28	0.16	**0.20**	0.10	0.23	0.14	0.20
LSTM(focal)	0.75	0.43	0.54	0.38	0.11	0.17	0.72	0.35	0.47	0.39	0.17	0.34	0.22	0.16	0.07	0.09	0.16	0.42	0.23	0.18
LSTM(F1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.22	0.49	**0.30**	0.06	**0.43**	0.11	0.08	0.44	0.13	0.18
CNN(cross-entropy)	0.91	0.34	0.49	**0.58**	0.08	0.14	0.65	0.36	0.46	0.36	0.27	0.23	0.25	0.25	0.01	0.02	0.19	0.17	0.18	0.15
CNN(focal)	**1.00**	0.28	0.44	0.44	0.14	0.22	0.73	0.28	0.40	0.35	**0.30**	0.30	**0.30**	0.13	0.11	0.12	0.17	0.37	0.23	0.22
CNN(F1)	0.73	0.39	0.50	0.43	0.19	**0.27**	0.73	0.40	0.52	**0.43**	0.18	0.50	0.26	0.10	0.23	0.14	0.12	**0.49**	0.19	0.20
RoBERTa(cross-entropy)	0.35	0.38	0.36	0.08	0.01	0.01	0.71	0.09	0.16	0.18	0.21	0.47	0.29	0.16	0.23	0.19	0.19	0.37	0.25	0.24
RoBERTa(focal)	0.36	0.27	0.31	0.00	0.00	0.00	0.35	0.08	0.12	0.14	0.24	0.41	**0.30**	0.15	0.20	0.17	0.16	0.31	0.21	0.23
RoBERTa(F1)	0.26	**0.63**	0.36	0.20	**0.22**	0.21	0.35	**0.51**	0.41	0.33	0.17	**0.63**	0.27	0.14	0.28	0.18	0.30	0.44	0.35	**0.27**
CNN-BERT(cross-entropy)	0.11	0.43	0.18	0.54	0.10	0.17	**0.96**	0.13	0.23	0.19	0.10	0.59	0.17	0.45	0.07	0.12	0.68	0.28	**0.39**	0.23
CNN-BERT(focal)	0.12	0.39	0.18	0.03	0.09	0.05	0.46	0.29	0.36	0.20	0.18	0.48	0.26	**0.63**	0.09	0.15	**0.76**	0.21	0.33	0.25
CNN-BERT(F1)	0.57	0.50	0.53	0.37	0.18	0.24	0.85	0.27	0.41	0.39	0.13	0.61	0.22	0.09	0.21	0.12	0.44	0.18	0.26	0.20

All the models were trained for 30 epochs. We stored models' checkpoints after each epoch and evaluated on the separate validation set. The models with lowest loss value on the validation were selected for the classification on the test set. The performance on the test set is shown in the table. Average F1 is the average of class-wise F1 scores.

We found that LSTM with the cross-entropy loss function achieved the highest F1 scores on identifying Issues and Conclusions which are, 0.58 and 0.53, respectively. Both CNN with the F1 loss function and CNN-BERT with cross-entropy loss have the highest F1 scores (0.27) on identifying Reasons. On average F1, CNN with the F1 loss function reached 0.43 which is highest among all the models.

The right side of of the table reports the performance of the models that were trained on the full text sentences corresponding to the automatically mapped human-annotated sentences from the summaries. Both LSTM with F1, CNN with focal and RoBERTa with focal tied in their performance on Issues (0.30). LSTM with cross-entropy has the highest F1 (0.20) on Reasons. For Conclusion classification, CNN-BERT with cross-entropy loss achieves the highest performance (0.39). Finally, RoBERTa with F1 loss achieves the highest score in terms of average F1.

Surprisingly, RoBERTa and CNN-BERT with cross-entropy and focal losses perform better on automatically mapped data than manually mapped data in terms of average F1. However, LSTM and CNN do not show the same pattern. The automatically mapped data are selected by sentence similarity scores with respect to BERT-Sentence embedding. RoBERTa and CNN-BERT somehow take the advantage of information contained in the sentence embedding to make a better classification. We are not sure how it affects the performance and will investigate it further. We also observed that models tend to perform better on Issue and Conclusion than Reason despite the type of training set. Since Reasons frequently include case facts, it is harder for models to classify them.

Despite the relative comparable F1 scores between training on manually mapped data and automatically mapped data, the precision of all types of sentences drops significantly in most cases when models are trained on automatically mapped data. This means that models have lower probability of making correct classifications.

When we compare the performance among the same models with different loss functions, the model with F1 loss function always has the highest average F1 score when trained on manually mapped data except for LSTM. The same pattern does not hold for the automatically mapped data. Each loss function has its own strength in terms of training set and model selection.

6 DISCUSSION AND ERROR ANALYSIS

6.1 Discussion

The results for classifying full text sentences trained on automatically mapped data, the right side of Table 2, are significantly higher than trained on annotated summaries in prior work. There the highest F1 scores for full texts trained on annotated summaries were Issue (0.27), Reason (0.14), and Conclusion (0.24). We attribute this improvement to using manually-mapped training sentences in the full texts, the higher numbers of annotated data, and the use of deep learning algorithms plus transformer models (LSTM, CNN, RoBERTa, and CNN-BERT).

As noted, we tried different kinds of neural models paired with different loss functions. We confirmed that the F1 loss function improved the performances of CNN and RoBERTa: RoBERTa with F1 loss yielded 0.21 on Reason and 0.41 on Conclusion while RoBERTa without F1 loss produced only 0.01 on Reason and 0.16 on Conclusion. When a loss function is aligned with the evaluation metric, it is likely to improve model performance. LSTM, however, did not perform well with the F1 loss function: on the manually mapped data , LSTM(F1) yielded 0.0 on all IRC types.

Those models each have advantages for certain sentence types. LSTM(cross-entropy) yielded the highest F1 scores on Issues and Conclusions. CNN(F1) and CNN-BERT(cross-entropy) performed best on identifying Reasons. In general, models have difficulty identifying Reason sentences, since Reasons have more complex semantic meanings. As noted, Reasons are intertwined with facts, which can easily be classified as the non-IRC type. The annotators

confirmed that Issues and Conclusions are easier to catch. They employ distinct keywords such as "issue", "conclusion", etc.

LSTM has the ability to detect temporal information about a sequence and can handle arbitrary input lengths. Meanwhile, CNN can only accept fixed size input. We think the ability to handle sequential information and longer lengths make LSTM more suitable for Issues and Conclusions, since these involve plainer language than Reasons. CNN has the upper hand on spotting Reasons. The literal composition of Reasons is more diverse than that of Issues and Conclusions; the convolutional features can capture this diversity.

6.2 Error Analysis

With respect to the right side of Table 2, the proof-of-concept study training models on automatically mapped data, the results suggest that classifying argument triples is feasible, but less effective than with the manually mapped data when taking precision and recall into account. We are particularly interested in the errors that the models made classifying Reasons. As noted, targeting the Reasons correctly is harder since they tend to be more complex and diverse than Issues and Conclusions.

Some of the misclassifications involved phrases attributing an expressed view to the judge. This is a positive sign, in that such self-referential judicial sentences are relatively less frequent in a case opinion and indicate sentences where the judge is more likely to assert that something is an Issue, Conclusion, or Reason. On the other hand, such self-referential attribution phrases do not necessarily discriminate among the three classifications.

7 FUTURE WORK

We plan to continue to annotate new cases in order to increase the size of the training set. Currently, the corpus includes 574 annotated summary / full text pairs. The size of the data set is still not large enough for adequately training more complex neural network models. The data set is sufficiently large, however, to allow us to continue to explore models and identify some challenges. The experience helps us to improve the quality of data as well as informs our intuitions about how human summarizers do their work. We expect that the more annotated data we collect the more interesting properties we will be able to observe in this process.

As noted, prior work explored different sampling strategies for dealing with imbalanced data to improve model performance. Different sampling methods have their merits in terms of their effects on training sets and model types. In this study, we briefly investigated a different method of adding augmented data to improve the performance of the models. Although the results were not as we expected, we observed that it had some positive effect on identifying Reasons from full texts. We will continue to explore other methods to deal with our imbalanced data.

We also plan to test whether a pre-trained legal language model improves performance over a generic language model.

ACKNOWLEDGMENTS

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] A. Bansal, Z. Bu, B. Mishra, S. Wang, K. Ashley, and M. Grabmair. 2016. Document Ranking with Citation Information and Oversampling Sentence Classification in the LUIMA Framework.
[2] J. Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1 (1960), 37–46.
[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[4] M. Falakmasir and K. Ashley. 2017. Utilizing Vector Space Models for Identifying Legal Factors from Text.. In *JURIX*. 183–192.
[5] A. Farzindar and G. Lapalme. 2004. Legal text summarization by exploration of the thematic structure and argumentative roles. In *Text Summarization Branches Out*. 27–34.
[6] V. Feng and G. Hirst. 2011. Classifying arguments by scheme. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. 987–996.
[7] C. Grover, B. Hachey, and C. Korycinski. 2003. Summarising legal texts: Sentential tense and argumentative roles. In *Proceedings of the HLT-NAACL 03 Text Summarization Workshop*. 33–40.
[8] Changai He, Sibao Chen, Shilei Huang, Jian Zhang, and Xiao Song. 2019. Using convolutional neural network with BERT for intent determination. In *2019 International Conference on Asian Language Processing (IALP)*. IEEE, 65–70.
[9] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. *CoRR* abs/1408.5882 (2014). arXiv:1408.5882 http://arxiv.org/abs/1408.5882
[10] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information* 10, 4 (2019), 150.
[11] J. Landis and G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics* (1977), 159–174.
[12] 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. *arXiv preprint arXiv:1907.11692* (2019).
[13] Qi. Lu, J. Conrad, K. Al-Kofahi, and W. Keenan. 2011. Legal document clustering with built-in topic segmentation. In *Proc. 20th ACM int'l conf. Info. and knowledge management*. 383–392.
[14] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao. 2020. Deep learning based text classification: A comprehensive review. *arXiv preprint arXiv:2004.03705* (2020).
[15] Amita Misra, Brian Ecker, and Marilyn A Walker. 2017. Measuring the similarity of sentential arguments in dialog. *arXiv preprint arXiv:1709.01887* (2017).
[16] R. Mochales and M. Moens. 2011. Argumentation mining. *Artificial Intelligence and Law* 19, 1 (2011), 1–22.
[17] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.
[18] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. http://arxiv.org/abs/1908.10084
[19] M. Saravanan and B. Ravindran. 2010. Identification of rhetorical roles for segmentation and summarization of a legal judgment. *Artificial Intelligence and Law* 18, 1 (2010), 45–76.
[20] J. Savelka and K. Ashley. 2018. Segmenting U.S. Court Decisions into Functional and Issue Specific Parts. In *Proceedings, 31st Int. Conf. on Legal Knowledge and Information Systems, Jurix*. 111–120.
[21] O. Shulayeva, A. Siddharthan, and A. Wyner. 2017. Recognizing cited facts and principles in legal judgements. *Artificial Intelligence and Law* 25, 1 (2017), 107–126.
[22] A. Wyner, R. Mochales-Palau, M. Moens, and D. Milward. 2010. Approaches to text mining arguments from legal cases. In *Semantic processing of legal texts*. Springer, 60–79.
[23] Huihui Xu, Jaromír Šavelka, and Kevin D Ashley. 2020. Using Argument Mining for Legal Text Summarization. *Legal Knowledge and Information Systems JURIX* (2020), 184–193.
[24] H. Yamada, S. Teufel, and T. Tokunaga. 2019. Building a corpus of legal argumentation in Japanese judgement documents: towards structure-based summarisation. *Art. Int. and Law* 27, 2 (2019), 141–170.