Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200862

Using Argument Mining for Legal Text Summarization

Huihui Xu [a,1], Jaromír Šavelka [b], and Kevin D. Ashley [a,c,d,2]
[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. Argument mining, a subfield of natural language processing and text mining, is a process of extracting argumentative text portions and identifying the role the selected texts play. Legal argument mining targets the argumentative parts of a legal text. In order to better understand how to apply legal argument mining as a step toward improving case summarization, we have assembled a sizeable set of cases and human-expert-prepared summaries annotated in terms of legal argument triples that capture the most important skeletal argument structures in a case. We report the results of applying multiple machine learning techniques to demonstrate and analyze the advantages and disadvantages of different methods to identify sentence components of these legal argument triples.

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

1. Introduction

Case summaries can assist legal professionals more easily to identify relevant cases and assess whether to read their full texts. As a more accessible means for the general public to gain some insights into what legal cases contain, summaries may also increase access to justice. A good case summary should include some key information: 1) major **issues** a court addressed in the case, 2) the court's **conclusion** with respect to each issue, and 3) a characterization of the court's **reasons** for reaching the conclusion. We refer to this key information as *legal argument triples* (*IRC triples*). These triples capture a skeletal structure of the legal arguments in a case.

Our ultimate goal is to extract the most important legal argument triples and use them to create succinct, three-sentence summaries that could enable legal researchers to better and more quickly assess what a case is really about and whether it is worth studying in detail. As a step in that direction, we conducted an empirical study of whether a machine learning (ML) model can identify the components of legal argument triples in case summaries prepared by human experts. The human summarizers are legal professionals charged with capturing the most important information in the cases. While their

[1] huihui.xu@pitt.edu
[2] ashley@pitt.edu

summaries are still too long for our intended information retrieval (IR) use case, they appear to contain the most important issues raised, the conclusions reached, and a reason connecting them. Since the experts act as a well-informed filter on importance, it makes sense to capture this information by annotating the summaries rather than the full texts. A more ambitious goal, however, is to generate the triples automatically from the full case texts.

Having developed a detailed annotation scheme, we annotated a sizable set of case summaries in terms of argument triples and also used those annotated summaries to help annotate the corresponding sentences in the full texts. We then applied various traditional ML algorithms and deep neural network models to identify the sentence components of IRC triples in corpora of legal summaries and of the corresponding full text decisions. We explored the use of different sampling strategies with the algorithms. We report the results and compare the advantages and disadvantages of the different methods.

2. Related Work

In order to summarize texts automatically, researchers have applied either abstractive [1,2] or extractive techniques. In summarizing legal texts, researchers have applied mainly the latter. See [3] for a recent survey. These extractive techniques have included: graph-based methods to cluster sentences by topics [4], by similarity based on repetition of legal phrases [5], or by unsupervised learning [6], machine learning classification of rhetorical roles of sentences in legal cases (e.g., FACT, BACKGROUND) [7,8,9], thematic structure [10,6], the rhetorical status of sentences in judgments of the UK House of Lords, [11], or catchphrases [12].

In [13] Zhong, et al. used machine learning to select which sentences in the decision are predictive of the case outcome. The summarizer computes the relative importance of sentences in a legal case document, as measured by their predictiveness and chooses a subset to generate the summary. They partitioned acceptable sentences as classified by type (i.e., Reasoning or Evidential Support sentence) and chose a set of summary sentences using maximum marginal relevance. They concluded, based on a detailed error analysis, that argument mining techniques would be required to identify more conceptual aspects of the decisions. Our focus on identifying legal argument triples is intended to do exactly that. In recent work, Yamada, et al. [14] have applied a similar approach to summarizing Japanese judgments by extracting issues, conclusions, and framings. Our legal argument triples, however, are simpler types, not tailored to Japanese legal judgements. Here, we provide evidence that machine learning can extract the argument triple cases from case summaries and full case texts based on a training set of expert summaries, a resource not available in the cited work.

Argument mining is "the automatic discovery of an argumentative text portion, and the identification of the relevant components of the argument presented there." [15]. Argument mining research has developed techniques to automatically identify argument components (e.g., premises, claims) in text and argumentative relations (e.g., support, attack) between components in contexts such as document summarization [16], legal information systems [17], and policy modeling platforms [18]. Argumentative relation mining involves determining if a relation holds among particular argument components and classifying the argumentative function of the relation (e.g., support vs. attack). Prior research

Table 1. Annotated data set summary.

			Sentence length (mean token count)			
	Count	# Sentences	Issue	Reason	Conclusion	non-IRC
Summaries	574	7,484	26.0	24.6	18.1	19.5
Full texts	109	23,653	35.5	26.7	25.7	16.8

has dealt with predicting argumentative relationship labels between pairs of argument components, e.g., attachment [19], support vs. non-support [20,21,22], {implicit, explicit}x{support, attack} [23,24] and verifiability of support [25]. In the legal domain, argument mining research has focused on extracting argumentative propositions, premises and conclusions, and nested arguments [26], arguments by example and other argument schemes [27], the rhetorical and other roles that sentences play in legal arguments [8,28], legal factors in domains like trade secret law [29], cited facts and principles (i.e., reasons or warrants) [30], functional and issue-related parts (including analysis and conclusions) [31], segments by topic [32] and segments by linguistic analysis [33,10,34].

3. Data Set

The Canadian Legal Information Institute (CanLII), a non-profit organization created and funded by the Federation of Law Societies of Canada[3] provided 28,733 paired cases and human-prepared summaries. The cases cover different kinds of legal claims and issues presented before Canadian courts. The summaries of those cases were prepared by members of Canadian legal societies.

Two annotators, both second year law students at the University of Pittsburgh, classified sentences from the summaries in terms of three types (i.e., issue, reason, conclusion), which together form "legal argument triples," and a catch-all category (for all the other sentences):

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.
4. **Non-IRC**– Sentences that do not qualify as either of the three types.

For annotation, we randomly selected 574 pairs from the 28,733 case/summary pairs. Annotators were asked to annotate all 574 summaries. After resolving all the annotation disagreements between annotators, we asked them to annotate the full texts corresponding to 109 summaries. Table 1 reports some key statistics about our annotated data set. The statistics of the mean sentence length reveal something of how the summaries are created. The IRC sentences are shorter in the summaries as compared to the corresponding sentences in the full texts. This likely reflects the fact that after selecting a sentence a human expert typically removes anything extraneous for the summary. The opposite holds for the non-IRC sentences, which suggests that the full texts have many short sentences not suitable for summaries (e.g., headings).

The third author, who is a law professor, provided a detailed annotation guideline for student annotators to identify sentences in the summaries that are instances of the

[3] CanLII's website is https://www.canlii.org/en/.

Table 2. Mean and median Cohen's kappa scores for each sentence type in summaries and full texts. Different degrees of agreement strength correspond to ranges of kappa: values ≤ 0.00 as poor agreement; value $0.00 - 0.20$ indicates slight agreement; value $0.21 - 0.40$ as fair agreement; value $0.41 - 0.60$ as moderate agreement; value $0.61 - 0.80$ as substantial agreement and value $0.81 - 1.00$ as almost perfect agreement.

	Summary				Full text			
	Issue	Reason	Concl.	Overall	Issue	Reason	Concl.	Overall
Mean κ	0.698	0.602	0.698	0.709	0.598	0.591	0.616	0.773
Median κ	1.000	0.700	1.000	0.740	0.780	0.820	0.750	0.860

three categories (i.e., issue, conclusion, and reason). Both annotators attended all the sessions. During those sessions, we did not notice any problems with American law school students working with Canadian cases.

The student annotators employed an online tool, Gloss (developed by the second author), to facilitate the annotation procedure. The annotators then proceeded over a period of several weeks to annotate successive batches of twelve case summaries at a time. After annotating each batch of twelve, the annotators and the first and third authors met via Zoom to resolve any differences and assign the final labels via consensus of both annotators in consultation with the third author.

The procedure for annotating full texts is different from that for summaries. We leveraged the existing summary annotations by allowing the student annotators to quickly target the sentences that are most similar to the annotated summary sentences: for each annotated sentence in a summary, annotators pick up some keywords and utilize them as pointers to locate corresponding sentences in the full text. This process does not require the annotators to read and understand the whole full texts and expedited the process of full text annotation. We believe that finding the triples in the summaries is considerably easier than doing so in the full texts. We hope to develop a strategy for inexpensively annotating full texts of cases that will enable us to amass a sizeable data set (something nonexistent in legal text summarization as of now). Eventually, we hope to project the annotations from the summaries to the full texts automatically. At the moment, however, this is done manually by the annotators. This paper could be understood as a first step toward the desired automation.

We use Cohen's κ [35] to measure the degree of agreement between the two annotators after their independent annotations of each batch of twelve summaries. They annotated N items into C mutually exclusive categories. In our case, there are three mutually exclusive categories — issue, conclusion and reason and the number of items are the number of sentences of each summary. The results of the inter-annotator agreement study are presented in Table 2. The mean Cohen's κ coefficients across all types is 0.716 which indicates a substantial agreement about the nature of the sentence types according to [36]. As shown in the table, the mean of Conclusion agreement scores is the highest whereas the mean of Reason's is the lowest. Reasons are clearly the most challenging since they are always entwined with facts. The other two types are easier because the courts are more explicit in identifying their Issues and the Conclusions. The feedback from the student annotators confirms this observation.

Figure 1 reports the distribution of the final consensus labels of summaries and full texts. Non-IRC is the most frequent label across all the summaries. The reason label is the second most frequent label, while the issue and conclusion labels are less frequent. This result is attuned to our intuition since more valuable sentences (IRC triples in our

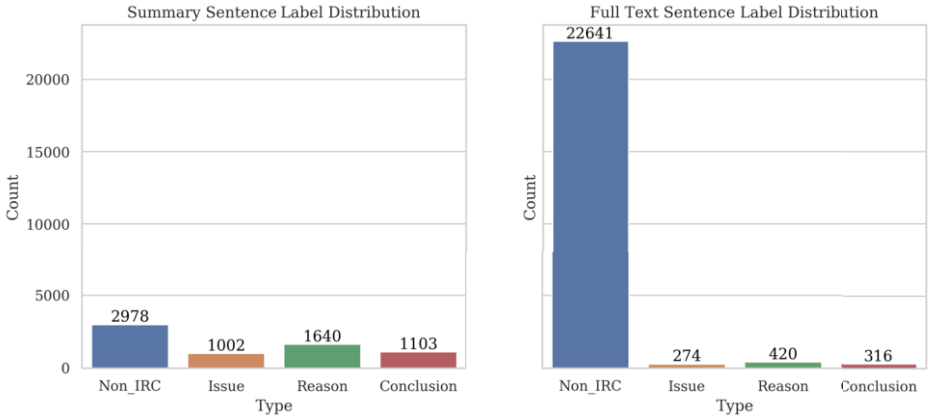

Figure 1. Distribution of annotated IRC type sentences in 574 summaries (left) and in 109 full texts (right). (The total number of full text IRC type sentences is lower because fewer full texts have been annotated than summaries.)

Table 3. Number of summaries that contain issue, reason and conclusion

	Issue	Reason	Conclusion
Count	557	531	574
Ratio	0.970	0.925	1.00

case) are much rarer than less valuable (Non-IRC) sentences. Table 3 reports the number of summaries that contain issue, reason and conclusion sentences. The statistics show that all summaries contain conclusions and well over 90% of summaries have issues (97%) and reasons (93%). This confirms our hypothesis that these sentence types are foundations for a good summary of a legal case.

4. Experiments

As discussed in Section 2, supervised learning techniques with labeled data are frequently used for argument mining. The performance of supervised learning techniques depends, among other things, on the quantity and quality of the annotated data. If the data set is too small, supervised learning algorithms will not have enough data to learn; if the quality of the annotation is not good, the algorithms will not be properly trained despite a large data set. As a result, a sizeable data set with consistent high quality annotation is fundamental for this study.

In order to fully assess the performance of our models, we undertook four types of experiments: Four-way classification on summary only (IRC types and the non-IRC type), four-way classification on full texts, binary classification on summaries (IRC vs non-IRC) and binary classification on full texts. Four-way classification on summaries takes annotated summaries as the training set and tests on unseen summaries. Four-way classification on full texts takes part of the annotated summaries as the training set and tests on full texts. Binary classification requires label transforming, which means we take annotated IRC labeled sentences as one group and the non-labeled sentences (Non-IRC)

as the other group. After transforming the labels, we use label-transformed summaries for training and testing on full texts. For four-way and binary summary classifications, we took 50% of our annotated summaries as the training set, 25% of them as the validation set and the rest 25% as the test set. Since the full texts are used as the test set, 75% of annotated summaries are being used as the training set and the rest of them are treated as the validation set. We carefully designed four-way and binary classification experiments on the full texts by excluding corresponding summaries from the training set to prevent leakage of test data into our training.

4.1. Traditional Machine Learning

From traditional ML we work with random forest as one of the most successful algorithms. A random forest algorithm (composed of multiple decision trees) was proposed in [37]. Instead of dealing with a single tree classifier, the random forest is an ensemble of trees and the final result depends on the majority vote. This algorithm significantly improves the classification accuracy because of the randomness of feature selection.

We use TF-IDF values of unigrams, bigrams and POS tags as features for the classifier. We utilize grid search to find the best parameters for this model. Grid search picks the parameters with the best validation accuracy. As [37] mentioned, there is a high probability that random forest will under perform in an extremely imbalanced data set since a bootstrap sample will contain only few or no instances of the minority classes. We observe that the non-IRC sentences are the majority in both summaries and full texts. Some research shows that down-sampling the majority class or over-sampling the minority class are effective ways to boost the performance of tree classifiers. We investigated different sampling strategies along with the random forest classifier and compared the final results: naive over- and under-sampling, performing over-sampling using synthetic minority over-sampling (SMOTE) and down-sampling by using edited nearest neighbor (ENN) [38], and SMOTE and down-sampling using TomekLinks [39].

4.2. Deep Neural Networks

Deep neural network techniques have been widely used for the text classification task because of their high performance. We leverage the power of deep learning to pick the argument triple components. We performed experiments with a Recurrent Neural Network (RNN) based model and a Convolutional Neural Network (CNN) based model.

RNN-based models take text as a sequence of words and are intended to capture the dependencies between words and text structures [40]. We use a variation of this RNN architecture—Long Short-Term Memory (LSTM) network. LSTM performs generally better than vanilla RNNs since LSTM addresses the vanishing gradient problem by introducing multiple gates to control the information flow into and out of the neural cells [40]. In our experiments, we use glove pre-trained word embedding. Those vectors were trained on 6 billion tokens and have 100 dimensions.

CNN-based models are often used for analyzing images. They utilize several filters to extract important features across several convolutional layers [41]. In text classification problems, a CNN model can use different filters looking at different word lengths in a piece of text. We use glove pre-trained word embedding for these experiments, as well.

4.3. FastText

In [42] a computationally efficient method for text classification is proposed. This model has only two layers, the embedding layer and linear layer. It has fewer parameters than most deep learning models. The embedding layer is used for calculating the word embedding, and taking the average of all the word embebddings. The average is stored in a variable and fed to the linear layer. Glove pre-trained word embedding is used for calculating the average in our experiments.

5. Results

The results of experiments described in Section 4 are presented in Tables 4 and 5. Table 4 reports the results of four-way classification on summaries and full texts. Here, CNN achieves the highest F1 scores on IRC types. FastText performs best on picking up issue and conclusion sentences in full texts. We found that the neural models perform better than the random forest model in terms of identifying components of legal argument triples in summaries. Our results also suggest that the different filters of a CNN model pick up more semantic cues regarding the sentence types than RNN models.

Table 5 reports the results of binary classification on summaries and full texts. For summary-only binary classification, we combined all the annotated IRC type sentences into one group. This significantly increases the ratio of majority and minority classes of our training set. Even though the random forest algorithm achieves some highest scores, the neural models and FastText have more stable performances than random forest for picking IRC type sentences in summaries: they all achieve 0.75 or above while the random forest model with naive random under-sampling, SMOTTENN, and SMOTETomek score less than 0.75. The only exception is random forest with oversampling technique.

Since full texts are significantly longer than summaries, the Non-IRC sentences are still significantly more numerous than IRC sentences even though we combine all three types of sentences. Training on an extremely imbalanced data set, random forest with different sampling techniques has a slight competitive edge over neural networks and FastText in classifying unseen samples. Random forest with sampling techniques score over 0.83 on Non-IRC recognition while neural models and FastText score less than 0.82. The performance of the neural models, however, is on par with random forest in picking IRC sentences. This result suggests that random forest may be a better choice for retrieving components of legal argument triples.

6. Discussion

We confirmed that classification techniques are able to extract the components of legal argument triples from summaries and full texts. We performed experiments using random forest and several deep neural network models. Those classifiers performed well on summary-only data. We observed that issue, conclusion, and non-IRC sentences are easier to classify correctly than reason sentences. This observation is aligned with the experience from the annotation phase: issue and conclusion sentences were easier for the human annotators to identify. This indicates that legal common knowledge is embedded

Table 4. F1 scores for the four-way classification on only-summary by using random forest (RF), LSTM, CNN and FastText. The weighted F1 is the average of F1 scores of each type weighted by its support. The suffixes are -O(over-sampling), -U(under-sampling), -w.o.r.(without replacement), - w.r.(with replacement).

	Issue		Reason		Conclusion		Non-IRC		Weighted F1	
	Sum.	Full	Sum.	Full	Sum.	Full	Sum.	Full	Sum.	Full
RF	0.48	0.19	0.39	0.10	0.58	0.23	0.67	**0.95**	0.56	**0.91**
RF-O	0.56	0.23	0.46	0.08	0.61	0.20	0.65	0.91	0.58	0.88
RF-U(w.o.r.)	0.55	0.15	0.48	0.07	0.59	0.15	0.58	0.80	0.55	0.77
RF-U(w.r.)	0.52	0.10	0.50	0.08	0.63	0.17	0.56	0.81	0.55	0.77
RF-SMOTEENN	0.49	0.14	0.09	0.08	0.58	0.21	0.66	0.92	0.48	0.89
RF-SMOTETomek	0.57	0.23	0.46	0.09	0.61	**0.24**	0.66	0.92	0.59	0.89
LSTM	0.59	0.13	0.52	0.09	**0.67**	0.17	0.68	0.85	0.62	0.82
CNN	**0.64**	0.23	**0.54**	0.10	**0.67**	0.20	0.66	0.85	**0.63**	0.82
FastText	0.59	**0.27**	0.52	**0.14**	**0.67**	0.22	**0.69**	0.89	**0.63**	0.86

Table 5. F1 scores for the binary classification on summaries and full texts by using random forest (RF), LSTM, CNN and FastText.

	IRC		Non-IRC		Weighted F1	
	Sum.	Full	Sum.	Full	Sum.	Full
RF	**0.76**	0.16	0.59	0.80	0.69	0.78
RF-O	**0.76**	0.15	0.59	0.83	0.70	0.80
RF-U(w.o.r.)	0.71	0.16	0.63	0.87	0.68	0.84
RF-U(w.r.)	0.70	0.17	0.64	0.92	0.67	0.89
RF-SMOTEENN	0.17	0.05	0.62	**0.98**	0.48	**0.94**
RF-SMOTETomek	0.75	0.16	0.63	0.80	0.70	0.77
LSTM	0.75	0.16	0.58	0.82	0.68	0.79
CNN	0.75	0.16	0.62	0.78	0.69	0.72
FastText	**0.76**	**0.18**	**0.65**	0.82	**0.72**	0.79

in the usage of semantic tokens and that classifiers can recognize them by training on a sizeable data set.

The more challenging task is migrating semantic cues of these sentence types to the broader context of full texts. Performance drops significantly when classifying sentences in full texts. One reason could be that the number of IRC sentences is still too few for training an ML classifier. We discovered that some sampling techniques helped to address the problem of imbalance; the combination of over-sampling and under-sampling techniques in SMOTETomek performed better than the others.

7. Conclusions and Future Work

We experimented with several ML models to identify components of legal argument triples by utilizing annotated human-generated summaries. We confirmed that classification techniques can extract components of these triples in both summaries and full texts. Based on the detailed discussion and evaluation, we found that neural models and FastText show promising results and some sampling techniques could be useful for boosting the performance of random forest.

In the future, we plan to increase the size of the annotated data set. The total number of case summaries is 574. We have used only 465 of them as a training set for our full text sentence classification because we needed to prevent our models from getting any cues from annotations of corresponding summaries. The data set supported our experiments in evaluating if ML techniques could identify components of legal argument triples and in recognizing challenges faced in this task. The data size, however, is still not large enough to draw finer conclusions in terms of comparing performance of different models where they reach similar levels of performance. A larger data set will also be helpful for testing on models that require careful hyperparameter tuning.

After improving a system's ability to identify sentence components of IRC triples, we will explore how best to identify related issues, conclusions, and reasons and to combine and present them as effective extractive case summaries.

Acknowledgement

Grants from the Autonomy through Cyberjustice Technologies Research Partnership at the University of Montreal Cyberjustice Laboratory supported this work. The Canadian Legal Information Institute provided the corpus of paired legal cases and summaries.

References

[1] K. Ganesan, Ch. Zhai, and J. Han. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. *Coling 2010*, 2010.

[2] A. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[3] P. Bhattacharya, K. Hiware, S. Rajgaria, N. Pochhi, K. Ghosh, and S. Ghosh. A comparative study of summarization algorithms applied to legal case judgments. In *Eur. Conf. on Info. Retrieval*, pages 413–428. Springer, 2019.

[4] M. Kim, Y. Xu, and R. Goebel. Summarization of legal texts with high cohesion and automatic compression rate. In *JSAI Int'l Symposium on Artificial Intelligence*, pages 190–204. Springer, 2012.

[5] F. Schilder and H. Molina-Salgado. Evaluating a summarizer for legal text with a large text collection. In *3rd Midwestern Computational Linguistics Colloquium (MCLC)*, 2006.

[6] M. Moens. Summarizing court decisions. *Info. processing & management*, 43(6):1748–1764, 2007.

[7] C. Grover, B. Hachey, I. Hughson, and C. Korycinski. Automatic summarisation of legal documents. In *Proceedings of the 9th international conference on Artificial intelligence and law*, pages 243–251, 2003.

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

[9] M. Yousfi-Monod, A. Farzindar, and G. Lapalme. Supervised machine learning for summarizing legal documents. In *Canadian Conference on Artificial Intelligence*, pages 51–62. Springer, 2010.

[10] A. Farzindar and G. Lapalme. Legal text summarization by exploration of the thematic structure and argumentative roles. In *Text Summarization Branches Out*, pages 27–34, 2004.

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

[12] Vu Tran, Minh Le Nguyen, Satoshi Tojo, and Ken Satoh. Encoded summarization: summarizing documents into continuous vector space for legal case retrieval. *Art. Int. and Law*, pages 1–27, 2020.

[13] L. Zhong, Z. Zhong, Z. Zhao, S. Wang, K. Ashley, and M. Grabmair. Automatic summarization of legal decisions using iterative masking of predictive sentences. In *Proc. 17th Int'l Conf. AI & Law*, pages 163–172, 2019.

[14] H. Yamada, S. Teufel, and T. Tokunaga. Building a corpus of legal argumentation in japanese judgement documents: towards structure-based summarisation. *Art. Int. and Law*, 27(2):141–170, 2019.

[15] A. Peldszus and M. Stede. From argument diagrams to argumentation mining in texts: A survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1):1–31, 2013.
[16] S. Teufel and M. Moens. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics*, 28(4):409–445, 2002.
[17] R. Mochales Palau and M. Moens. Argumentation mining: the detection, classification and structure of arguments in text. In *Proc. 12th int'l conference on artificial intelligence and law*, pages 98–107, 2009.
[18] E. Florou, S. Konstantopoulos, A. Koukourikos, and P. Karampiperis. Argument extraction for supporting public policy formulation. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 49–54, 2013.
[19] A. Peldszus and M. Stede. Joint prediction in mst-style discourse parsing for argumentation mining. In *Proc. 2015 Conf. on Empirical Methods in Natural Language Processing*, pages 938–948, 2015.
[20] O. Biran and O. Rambow. Identifying justifications in written dialogs by classifying text as argumentative. *International Journal of Semantic Computing*, 5(04):363–381, 2011.
[21] E. Cabrio and S. Villata. Combining textual entailment and argumentation theory for supporting online debates interactions. In *Proc. 50th Ann. Mtg. Assoc. for Comp. Ling. (Vol. 2)*, pages 208–212, 2012.
[22] C. Stab and I. Gurevych. Identifying argumentative discourse structures in persuasive essays. In *Proc. 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56, 2014.
[23] F. Boltužić and J. Šnajder. Back up your stance: Recognizing arguments in online discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58, 2014.
[24] H. Nguyen and D. Litman. Context-aware argumentative relation mining. In *Proc. 54th Ann. Mtg. of the Assoc. for Comp. Linguistics (Vol. 1)*, pages 1127–1137, 2016.
[25] J. Park and C. Cardie. Identifying appropriate support for propositions in online user comments. In *Proceedings of the first workshop on argumentation mining*, pages 29–38, 2014.
[26] R. Mochales and M. Moens. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22, 2011.
[27] V. Feng and G. Hirst. Classifying arguments by scheme. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 987–996, 2011.
[28] A. Bansal, Z. Bu, B. Mishra, S. Wang, K. Ashley, and M. Grabmair. Document ranking with citation information and oversampling sentence classification in the luima framework, 2016.
[29] M. Falakmasir and K. Ashley. Utilizing vector space models for identifying legal factors from text. In *JURIX*, pages 183–192, 2017.
[30] O. Shulayeva, A. Siddharthan, and A. Wyner. Recognizing cited facts and principles in legal judgements. *Artificial Intelligence and Law*, 25(1):107–126, 2017.
[31] J. Savelka and K. Ashley. Segmenting u.s. court decisions into functional and issue specific parts. In *Proceedings, 31st Int. Conf. on Legal Knowledge and Information Systems, Jurix*, pages 111–120, 2018.
[32] Qi. Lu, J. Conrad, K. Al-Kofahi, and W. Keenan. Legal document clustering with built-in topic segmentation. In *Proc. 20th ACM int'l conf. Info. and knowledge management*, pages 383–392, 2011.
[33] C. Grover, B. Hachey, and C. Korycinski. Summarising legal texts: Sentential tense and argumentative roles. In *Proceedings of the HLT-NAACL 03 Text Summarization Workshop*, pages 33–40, 2003.
[34] A. Wyner, R. Mochales-Palau, M. Moens, and D. Milward. Approaches to text mining arguments from legal cases. In *Semantic processing of legal texts*, pages 60–79. Springer, 2010.
[35] J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.
[36] J. Landis and G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174, 1977.
[37] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
[38] G. Batista, R. Prati, and M. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.
[39] G. Batista, A. Bazzan, and M. Monard. Balancing training data for automated annotation of keywords: a case study. In *WOB*, pages 10–18, 2003.
[40] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao. Deep learning based text classification: A comprehensive review. *arXiv preprint arXiv:2004.03705*, 2020.
[41] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.
[42] A. Joulin, E. Grave, and T. Bojanowski, P.and Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.