Legal Knowledge and Information Systems
M. Palmirani (Ed.)

doi:10.3233/978-1-61499-935-5-111

Segmenting U.S. Court Decisions into Functional and Issue Specific Parts

Jaromír ŠAVELKA [a,b,1] and Kevin D. ASHLEY [a,b,c]
[a] *Intelligent Systems Program, University of Pittsburgh, USA*
[b] *Learning Research and Development Center, University of Pittsburgh, USA*
[c] *School of Law, University of Pittsburgh, USA*

Abstract. In common law jurisdictions, legal research often involves an analysis of relevant case law. Court opinions comprise several high-level parts with different functions. A statement's membership in one of the parts is a key factor influencing how the statement should be understood. In this paper we present a number of experiments in automatically segmenting court opinions into the functional and the issue specific parts. We defined a set of seven types including Background, Analysis, and Conclusions. We used the types to annotate a sizable corpus of US trade secret and cyber crime decisions. We used the data set to investigate the feasibility of recognizing the parts automatically. The proposed framework based on conditional random fields proved to be very promising in this respect. To support research in automatic case law analysis we plan to release the data set to the public.

Keywords. case law, legal analysis, information retrieval, text segmentation, conditional random fields

1. Introduction

In this paper we examine an application of natural language processing (NLP) and machine learning (ML) to facilitate one of the initial steps in case law analysis. Court opinions consist of several high-level parts each of which has a different function. The main Analysis part often contains several sub-parts each of which is dedicated to a different issue. Distinguishing the functional as well as issue-specific parts is crucial for a lawyer to be able to focus attention on the pieces of the opinion that matter. We are interested if and how could NLP and ML techniques be helpful in recognition of the individual parts. We address this question by assessing the ability of the presented NLP/ML pipeline to perform this step in the analysis automatically.

2. Background and Motivation

We define the task as a two step process (diagrammed in Figure 1). In the first step an opinion is segmented into a varying number of consecutive non-overlapping parts. Each part is assigned one of the following types:

[1]Corresponding Author: Jaromír Šavelka, Learning Research and Development Center, 3939 O'Hara St, Office 520, Pittsburgh, PA 15260, USA; E-mail: jas438@pitt.edu.

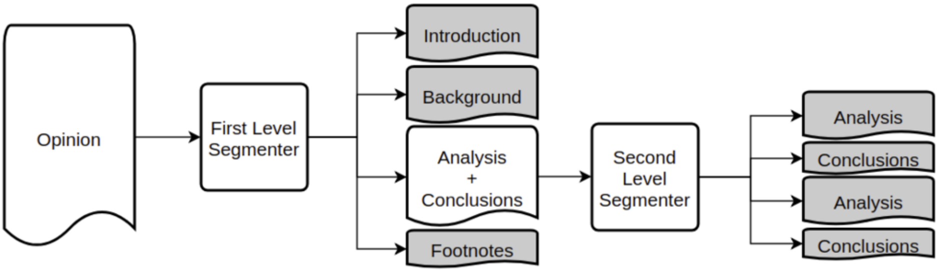

Figure 1. The diagram shows the two-step segmentation process and its interim and final outcomes.

1. **Introduction** – the opening part which typically consists of lines indicating the deciding court, judges, the case citation, parties, etc. It is not uncommon that the court would include a summary of the decision.
2. **Background** – the part where the court describes the procedural history of the case, the relevant facts, as well as what the parties are claiming. Its tone is usually descriptive, i.e., the court refrains from expressing its own opinions.
3. **Analysis** – the part where the court discusses and reasons about the issues of the case and states its outcome. Quite often the tone is deliberative, i.e., the court expresses opinions on the issues, arguments, or claims. The court may deal with a single issue or it may treat several issues separately.
4. **Concurrence or Dissent** – the part where opinions of concurring or dissenting judges are presented. There may be dedicated sections for each concurrence/dissent but there could also be just a single sentence informing about the list of concurring/dissenting judges.
5. **Footnotes** – a list the indices of which are references to different parts of a decision. Each item of the list provides additional information as to what is in the text at the place of reference.
6. **Appendix** – a separate document attached to a decision as supplemental material.

The type system is a variation on type systems presented by a number of authors in the past (see Section 3 on related work). It is specifically tailored for court decisions in the U.S. In our opinion it would generalize well to other jurisdictions. Note that the task is more complex than merely identifying sections or section headings. Most of the time the decisions are not explicitly segmented into sections that would map onto the scheme.

In the second step the Analysis part, that typically spans the larger portion of an opinion, is annotated with the one additional type:

7. **Conclusions** – the part where a court states the outcome of its analyses, i.e., its decision concerning each issue it addresses.

The annotations are then used to segment the Analysis part into the treatments of specific issues. Although, the situation is more complex than that, one could adopt an assumption that argumentation about a specific issue is finished with its outcome (Conclusions). The segments could then be obtained in a straightforward way.

This line of work has been recognized as "important but often neglected" in [25] where the author speaks about segmentation of a legal document into its structural elements such as, e.g., facts, arguments, and rulings. As explained in [15], "The ability to identify and partition a document into segments is important for many NLP tasks, including information retrieval, summarization, and text understanding." Mainly, this is because the segments provides clues to the meaning of their contents [1].

For instance, knowing in which high-level part a sentence appears will help to annotate sentences in terms of the roles they play in legal argument. Sentences that state a finding of fact or state a legal rule are more likely to be found in the Background or Analysis parts, respectively. Annotating Conclusions will help to indicate where the analysis of one issue ends and another begins. We plan to use document segmentation as the next step in our long-term project of automatically analyzing the texts of court decisions to support statutory interpretation [21,22,23].

The annotation task also may have important pedagogical potential. While it is easy for law student annotators to identify instances of Background, Concurrence or Dissent, and Footnotes, it is more difficult to identify a court's conclusions regarding the issues raised. Student annotators could benefit from this kind of practice while providing training instances for machine learning and legal text analytics.

3. Related and Prior Work

Unlike our project, much of the related work on segmenting texts into multi-paragraph passages focuses on dividing the segments by topic (e.g., [14] and [15]). Some domain general approaches to segmenting texts into multi-paragraph passages by topic are based on statistical similarity and lexical cohesion, the repetition of similar words in coherent segments and the tendency for vocabulary to change across segment boundaries (e.g., [4] and [12]). Segmenting legal texts into topics or, as in our project, into functional sections or parts, has required the application of more legal domain-specific knowledge [7,26]. For instance, one must first settle on the types of functional sections that are present in the legal texts of interest such as courts' legal decisions. In [5], based on discussions with a focus group of West editors and an exercise in which each editor tagged 2-3 cases, the authors developed a short list of segment types which could be linked to fixed issues or annotations: Issues / Contentions (Substantive and Procedural) Analysis, Facts/Evidence, and Conclusions (Abstract and Concrete).

One approach to segmentation has focused on automatically identifying the rhetorical roles of sentences. For instance, a case document to be summarized has been divided into parts for purposes of selecting the important sentences and organizing them into a summary based on a standard model of case structure [7,10,16]. Supervised ML has been applied to learn rhetorical role sentence classifiers based on a wide range of features [10,18]. Unsupervised ML was applied to select relevant portions of texts for summaries [16]. [16]

Other work has focused more specifically on linguistic analysis of sectional texts to identify features characteristic of section types. The authors of [9] employed verb tense and aspect in sentences stating legal background knowledge, case description, or a judge's opinions. [3] employed other linguistic markers with contextual dependencies to construct a thematic structuring rule base for contextual exploration. In [7] the authors employed linguistic markers to segment Canadian decisions into four units: Introduction, Context, Juridical Analysis, and Conclusion. The first three unit types appear to map onto our Introduction, Background, and Analysis. The Conclusion appears to correspond to the "Final decision of the court," whereas we treat it more broadly; where the court treats several issues separately there may be a conclusion at the end of the Analysis of each issue. A similar scheme was proposed in [11], including some additional types such as Dissent, Footnotes, or Party Claims. Identifying Conclusions in our work appears to be related to some aspects of the work presented in [27]. The authors identify typical language structures that are used in various types of Premises or Conclusions. These are then expressed in the form of Context Free Grammar for parsing legal arguments. Some of the types used in the pipeline presented in [8] and [2] appear to partially map to our Conclusions as well.

By contrast to the above work, however, we have not employed rhetorical roles or linguistic analysis in our project.

In [19] conditional random fields (CRF) were applied to segment legal documents into seven labeled components with each label representing a corresponding rhetorical role. CRF were applied to identify the rhetorical roles: identifying case (F_1 .853), establishing case facts (F_1 .824), arguing case (F_1 .805), case history (F_1 .851), arguments (F_1 .787), decision ratio (F_1 .888), final decision (F_1 .973). Features included key phrases, named entities recognized, proper names, location in layout, and legal vocabulary, neighboring sentence similarity, paragraph structure, and citation [18].

The authors of [26] applied other ML algorithms (naive Bayes, logistic regression, decision trees, support vector machines and neural networks) to identify sections but only of legal briefs and considering only section titles. The authors in [24] employed a naive Bayes multinomial classifier to automatically annotate legal principles in case texts based on features including deontic modalities of verbs such as must, may, or should.

We employ a corpus containing considerably more legal decisions than in the above work and covering a wider range of legal domains. We also use the sentence boundary detection system developed in [20] and [22]. This is a crucial component that allows us to use sentence-level segments and sentence-level features in the presented NLP/ML pipeline.

4. Experimental Design

4.1. Data Sets

We downloaded 316 court decisions from the online Court Listener[2] and Google Scholar[3] services. Of these 143 are from the area of cyber crime (cyber bullying, credit card frauds, possession of electronic child pornography), and 173 cases involve trade secrets (typically misappropriations). The trade secret part of the corpus is a slightly extended data set assembled in [6]. We use cases from the two different areas of law to gain a sense of how well the trained models generalize (see Section 6 for details).

We created guidelines for manual annotation of the decisions[4] with the types introduced in Section 2. Two human annotators (the authors) then annotated the decisions using Gloss, the web-based annotation environment developed by the authors. Each decision was annotated by one of the annotators. A subset (25 from each domain, i.e., 50

[2] `www.courtlistener.com`
[3] `scholar.google.com`
[4] Available at `https://github.com/jsavelka/us-dec-func-iss-sgm.git.`

	Cyber Crime			Trade Secrets			Total		
	Filtered	Original	Agree	Filtered	Original	Agree	Filtered	Original	Agree
Documents	139	143	–	158	173	–	297	316	–
Introduction	139	143	.965	158	173	.934	297	316	.947
Background	133	139	.758	152	172	.787	285	311	.774
Analysis	139	143	.932	158	173	.936	139	143	.935
Conclusions	398	429	.689	833	909	.763	398	1338	.732
Concurr/Diss	0	18	1.0	0	44	1.0	0	62	1.0
Footnotes	95	97	.983	103	113	.982	198	210	.982
Appendix	0	7	–	0	6	–	0	13	–

Table 1. The data sets before (Original) and after processing (Filtered) with inter-annotator agreement (Agree).

in total) was annotated by both authors to measure inter-annotator agreement (see Table 1). The inter-annotator agreement is computed using the following formula:

$$A = \sum_{a=0}^{1} \sum_{i=1}^{|S|} \frac{t(s_i, a) = T \wedge t(s_i, \neg a) = T}{t(s_i, a) = T}$$

In the formulas S stands for the set of all sentences in the corpus; T means a specific type; $t(s_i, 0)$ stands for the type of the sentence s_i assigned by the first human annotator; $t(s_i, 1)$ those assigned by the second human annotator; $\neg$ is a negation that can be applied only to 0 or 1 in order to reverse them into each other. We use the accuracy formula because it maps nicely to the F_1-measure that we use for the evaluation of the proposed framework. This allows us to assess how well the machine performs when compared to a human.

Table 1 provides detailed statistics of the created annotations. Certain pre-processing steps involving exclusion of some documents as well as parts of others were necessary. We report the statistics of the full data set in the "Original" columns of Table 1; statistics of the data set after pre-processing are reported in the "Filtered" columns. These were performed in order to keep the experiments reported in this paper focused on the main task. A small number of decisions have a non-standard structure where, for example, Background interleaves with Analysis and Conclusions. Since we do not have enough data in our data set to deal with these, we decided to exclude the documents from our experiments. In addition, it turned out that we have very little data for Appendix and Concurrence and Dissent types. We manually eliminated them out for the purposes of the experiments reported here.

4.2. *Classification Pipeline*

The classification pipeline is schematically depicted in Figure 2. Each document is first split into individual paragraphs. Since the opinions are relatively clean a simple regular expression was used to perform this step. Information about the document from which a sentence comes and its order in a sequence are retained. The paragraphs are then split into individual sentences using the sentence boundary detection system of [20].

The paragraphs and sentences are transformed into vectors of paragraph level features. The features include, e.g., lower-case tokens and POS-tagged lemmas that appear in a paragraph, the position of a paragraph within a document, its length as well as the average length of sentences it contains. The first and the last five tokens (paragraph boundaries) are described through more detailed features, such as a token's signature, length, and type (i.e., digit, case, white space).

The resulting feature vectors and human created annotations (as labels) are then used in the training of the Functional parts segmenter (the first step of the analysis). The segmenter consists of three CRF models. A CRF is a random field model that is globally conditioned on an observation sequence O. The states of the model correspond to event labels E. We use a first-order CRF in our experiments (observation O_i is associated with E_i). We use the CRFsuite[5] implementation of a first-order CRF [13,17].

The three models are trained in an iterative manner. First, the model for recognizing the Introduction type is trained on full texts. Then we train a model for separating the Background type from the rest in documents that were stripped of the Introduction parts. Finally, a model for finding the boundary between the Analysis and Footnotes is trained on the documents that were stripped of the Introduction and Background types.

The sentences from the Analysis part are transformed into vectors of sentence level features. The types of features are similar to those described above except they are applied at a different level (sentence as opposed to paragraph). The feature vectors and human-created annotations are then used in the training of the Conclusions recognizer (the second step of the analysis). The recognizer consist of the single CRF model.

The two resulting components—the Functional parts segmenter and the Conclusions recognizer—are then used in the automated annotation process on the unseen document. This process is similar to the one described above. Where the models required human-created annotations during training, the automatically generated annotations are used. This happens in the interim stages of the Functional parts segmenter and when feeding the (predicted) Analysis part to the Conclusions recognizer.

4.3. Evaluation

We use 10-fold cross validation as the evaluation method. This means that the data set is split into 10 folds of roughly equal sizes. The splitting is performed at the level of documents, i.e., all the sentences from a single document are in the same fold. We did not consider the size of a document for the purposes of splitting. Therefore, the number of sentences in each fold could vary significantly. We evaluate performance on each fold separately based on the model trained on the other 9 folds. The reported results are an aggregate from all the 10 folds.

We use precision (P), recall (R), and F_1-measure (F_1), i.e., the traditional information retrieval measures, to evaluate performance of the presented pipeline. The performance is evaluated at the level of sentences where the type of the reported measures is micro. Therefore, measures are computed as follows:

$$P = \sum_{i=1}^{|S|} \frac{t(s_i, g) = T \wedge t(s_i, p) = T}{t(s_i, p) = T} \qquad R = \sum_{i=1}^{|S|} \frac{t(s_i, g) = T \wedge t(s_i, p) = T}{t(s_i, g) = T} \qquad F_1 = \frac{2PR}{P+R}$$

[5] www.chokkan.org/software/crfsuite

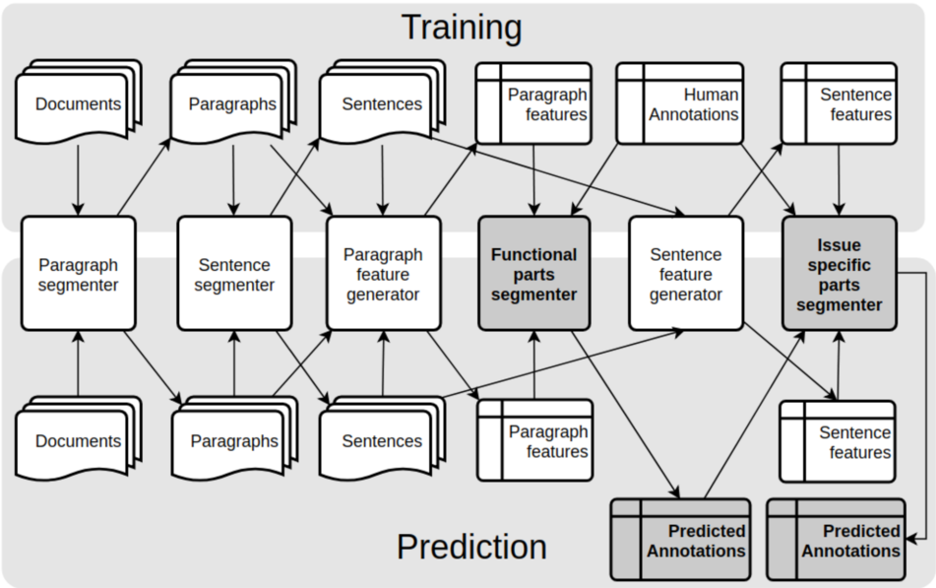

Figure 2. The diagram shows the automatic classification pipeline (training and prediction).

	Cyber Crime			Trade Secrets			Total		
	P	R	F_1	P	R	F_1	P	R	F_1
Introduction	.926	.945	.935	.907	.947	.926	.914	.947	.930
Background	.634	.752	.689	.640	.775	.701	.638	.767	.697
Analysis	.922	.879	.900	.948	.871	.908	.939	.873	.905
Footnotes	.874	.963	.916	.845	.981	.908	.855	.973	.910

Table 2. Results of the segmentation into the high-level functional parts.

In the formulas S stands for the set of all sentences in the corpus; T means a specific type; $t(s_i, g)$ stands for the type of the sentence s_i assigned by a human annotator; $t(s_i, p)$ that assigned by the automatic pipeline.

5. Results

Table 2 summarizes the results of the experiments evaluating the feasibility of the first step described in Section 2, i.e., the segmentation of an opinion into high-level functional parts. The performance of the models differs considerably across the types but it correlates well across the two different domains as well as with the inter-annotator agreement. The recognition of the Introduction and the Analysis types has a high success rate. It appears to be comparable to the human performance. The performance on the Background and the Footnotes types is slightly lower but still very close to the human performance measured through the inter-annotator agreement (see Table 1). Due to data sparsity we did not attempt to predict the Appendix and the Concurrence and Dissent types.

Table 3 presents the results of the experiments evaluating the feasibility of the second step described in Section 2, i.e., annotation of the Analysis part with the Conclusions

	Cyber Crime			Trade Secrets			Total		
	P	R	F_1	P	R	F_1	P	R	F_1
Conclusions	.521	.461	.489	.576	.493	.532	.552	.482	.515

Table 3. Results of the Conclusions recognition.

type. Although, the performance is promising the experiments confirm that this task is very challenging. From the inter-annotator agreement it appears that this task is the most challenging one. We further elaborate on the results in Section 6.

6. Discussion

We have detected a problem of data sparsity pertaining to certain types. Whereas types such as the Introduction or the Analysis are nearly guaranteed to appear in each single document, types such as the Appendix or the Concurrence and Dissent are present only occasionally. In order to solve this problem one would have to enrich the data set with many opinions that specifically contain these parts. However, this will necessarily lead to a biased data set. The result may be a model that is over-predicting the rare types. On the other hand, unbiased sampling could require a corpus that would be prohibitively expensive (in terms of annotation labor).

We performed a detailed error analysis on the Conclusions recognition step. The sentences that were identified by the human annotator as Conclusions but missed by the system were often quite short (e.g., "There is no error."). Typically, these sentences consist of words that are quite common and are not too informative for a system to pick up the signal. Some of the shorter length sentences were recognized correctly (e.g., "The judgment of the district court is affirmed.").

With respect to the sentences that the system erroneously predicted as the Conclusions it appears that attribution is a major challenge. Some sentences included verbs like "conclude" or "hold" which are likely very suggestive for a sentence to be classified as the Conclusions. However, the sentence was not attributed to the deciding court. Instead it was attributed to some other entity, such as a lower court or a party.

Words such as "therefore" or various forms of "find" (i.e., "finds" or "finding") may often indicate the Conclusion type. However, they also appear in many sentences that are not the Conclusions of issues, for instance, "Therefore, we now address appellants' claims," "finds that defendant's reliance [on certain cases] is misplaced," "hereby finds the following facts and state separately its conclusions of law," or "because there was no jury finding against her."

The error analysis also confirmed the task is very challenging even for humans. From the trade secrets case we sampled 40 sentences that were predicted as the Conclusions by the system, but not annotated as such by the human annotator. In case of 15 of those the human annotator would be willing to change the original decision. This is tightly connected to the problem of ambiguity as to what constitutes the conclusion of an issue. For example, when are findings of fact conclusions as to an issue? In addition, a court may break an issue down into multiple sub-issues like whether a rule applies (e.g., "We conclude that this rule applies equally to both blueprints and/or drawings and customer lists ...") or whether evidence supports a conclusion (e.g., "We find that X is further evidence of Y", "We credit that testimony.")

7. Future Work

The models trained in our experiments operate on low-level textual features. While examining the errors it became clear that while these could be sufficient for certain tasks (e.g., the segmentation into functional parts) they might be insufficient for others (e.g., recognizing Conclusions). We already pointed out that an attribution resolution would likely improve the Conclusions detection.

One of the problems that we detected is data sparsity in case of certain types. The challenge is how to obtain a significantly larger set of annotated data. We hypothesize that law students can annotate legal texts as a useful pedagogical exercise and that their annotations could then be used for purposes of ML and legal text analytics. In a small pilot study, ten students in Ashley's Intellectual Property course employed Gloss to annotate the parts of the full texts of four trade secret law cases before reading the edited versions in the case book. They received a 15-page set of "Instructions for Annotating Cases Using Gloss comprising a 2-page guide to access Gloss and use it to annotate the cases, the definitions of the seven parts as above, specific guidelines for annotating each part, and multiple samples illustrating the annotations and the borders between parts. Across the four cases, the students made nearly 1000 annotations.

Given this preliminary evidence that law students can use Gloss, in future work, we plan to arrange for students to annotate legal decisions in terms of key aspects of the reasoning in a case beyond high-level parts and conclusions, including stating a legal rule, expressing a judge's holding that a rule requirement is satisfied (or not), reporting a finding of fact, describing evidence, and substantive features of legal domains such as legal factors, patterns of fact that strengthen or weaken a sides position on a claim. We also plan to evaluate if and how much they learn by testing students' knowledge gains and by monitoring any increases in the extent to which their annotations agree with an instructor's and each other's.

8. Conclusions

In this paper we examined the possibility of automatically segmenting court opinions into high-level functional (step 1) and issue specific (step 2) parts. We have shown that segmentation into the functional parts could be done automatically in a quality that is not too far from human performance. Although, the model for segmenting the Analysis part into issue-specific segments via the Conclusions recognizer shows promise, there appears to be a gap between its performance and that of a human annotator. We hope that this work will stimulate further research in segmentation of court opinions. For this reason we will release the data set we employed in these experiments.

Acknowledgements

This work was supported in part by the National Institute of Justice Graduate Student Fellowship (Fellow: Jaromir Savelka) Award # 2016-R2-CX-0010, "Recommendation System for Statutory Interpretation in Cybercrime."

References

[1] Ashley, K. *Artificial intelligence and legal analytics: new tools for law practice in the digital age*. Cambridge University Press, 2017.

[2] 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." *JURIX*, v. 294, p. 33. IOS Press, 2016.

[3] Berrazega, I., and Faiz, R. "Automatic structuring of Arabic normative texts." *Proceedings of the Computational Methods in Systems and Software* 238-249. Springer, 2017.

[4] Choi, F. Y., Wiemer-Hastings, P., and Moore, J. "Latent semantic analysis for text segmentation". *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, 2001.

[5] Conrad, J. and Dabney, D. "A cognitive approach to judicial opinion structure: applying domain expertise to component analysis." *Proc. 8th Int'l Conf. on Artificial intelligence and Law* 1-11. ACM, 2001.

[6] Falakmasir, M.H. and Ashley, K.D. "Utilizing Vector Space Models for Identifying Legal Factors from Text." *JURIX* 183-192, 2017.

[7] A. Farzindar and G. Lapalme. "Letsum, an automatic legal text summarizing system." *Legal knowledge and information systems, JURIX* 11-18, 2004.

[8] Grabmair, M., K. Ashley, R. Chen, P. Sureshkumar, C. Wang, E. Nyberg, and V. Walker. "Introducing LUIMA: an experiment in legal conceptual retrieval of vaccine injury decisions using a UIMA type system and tools." In *Proc. 15th Int'l Conf. on Artificial Intelligence and Law*, pp. 69-78. ACM, 2015.

[9] Grover, C., Hachey, B., Hughson, I., and Korycinski, C. "Automatic summarisation of legal documents." *Proc. 9th Int'l Conf. on Artificial intelligence and law* 243-251. ACM. 2003.

[10] B. Hachey and C. Grover. "Extractive summarisation of legal texts." *Artificial Intelligence and Law* 14,4 305-345, 2006.

[11] Harasta, J., F. Kasl, J. Misek, and J. Savelka. "Segmentation of Czech Court Decisions into Subtopic Passages." *CEILI Workshop on Legal Data Analysis* JURIX, 2017.

[12] Hearst, M. "TextTiling: Segmenting text into multi-paragraph subtopic passages." *Computational Linguistics* 23.1, 33-64. 1997.

[13] Lafferty, J., A. McCallum, and F. Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." 2001.

[14] Le, T. T. N. "A study on Hierarchical Table of Indexes for Multi-documents." *Ph.D. Diss., Japan Advanced Inst. of Science and Technology* 1-47. 1999.

[15] Lu, Q., Conrad, J., Al-Kofahi, K., and Keenan, W. "Legal document clustering with built-in topic segmentation." *Proc. 20th ACM Int'l Conf. on Information and Knowledge Management* 383-392. ACM. 2011.

[16] M.-F. Moens. "Summarizing court decisions." *Information Processing and Management* 43, 6, 1748-1764. 2007.

[17] Okazaki, N. "Crfsuite: a fast implementation of conditional random fields." 2007.

[18] Saravanan, M., and B. Ravindran. "Identification of rhetorical roles for segmentation and summarization of a legal judgment." *Artificial Intelligence and Law* 18.1 45-76. 2010.

[19] Saravanan, M., B. Ravindran, and S. Raman. "Improving legal document summarization using graphical models." *JURIX* 2006.

[20] Savelka, J., V. Walker, M. Grabmair, and K. Ashley. "Sentence Boundary Detection in Adjudicatory Decisions in the United States." *Traitement Automatique des Langues* 58.2 21-45. 2017.

[21] Savelka, J., and K. Ashley. "Detecting Agent Mentions in US Court Decisions." *JURIX*. 2017.

[22] Savelka, J., and K. Ashley. "Using Conditional Random Fields to Detect Different Functional Types of Content in Decisions of United States Courts with Example Application to Sentence Boundary Detection." *ASAIL 2017: 2nd Workshop on Automated Semantic Analysis of Information in Legal Texts*.

[23] Savelka, J., and K. Ashley. "Extracting case law sentences for argumentation about the meaning of statutory terms." *Proc. of the Third Workshop on Argument Mining (ArgMining2016)*. 2016.

[24] Shulayeva, O., Siddharthan, A., and Wyner, A. "Recognizing cited facts and principles in legal judgements." *Artificial Intelligence and Law* 25(1),107-126. 2017.

[25] Schweighofer, E. "The Role of AI and Law in Legal Data Science." *JURIX* 191-192. 2018.

[26] Vanderbeck, S., Bockhorst, J., and Oldfather, C. "A Machine Learning Approach to Identifying Sections in Legal Briefs." *MAICS* 16-22. 2011.

[27] Wyner, A., Mochales-Palau, R., Moens, M. F., & Milward, D. Approaches to text mining arguments from legal cases. In *Semantic processing of legal texts* (pp. 60–79). Springer, Berlin, Heidelberg. 2010.