Legal Knowledge and Information Systems
A. Wyner and G. Casini (Eds.)

doi:10.3233/978-1-61499-838-9-39

Detecting Agent Mentions in U.S. Court Decisions

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

Abstract. Case law analysis is a significant component of research on almost any legal issue and understanding which agents are involved and mentioned in a decision is integral part of the analysis. In this paper we present a first experiment in detecting mentions of different agents in court decisions automatically. We defined a light-weight and easily extensible hierarchy of agents that play important roles in the decisions. We used the types from the hierarchy to annotate a corpus of US court decisions. The resulting data set enabled us to test the hypothesis that the mentions of agents in the decisions could be detected automatically. Conditional random fields models trained on the data set were shown to be very promising in this respect. To support research in automatic case-law analysis we release the agent mentions data set with this paper.

Keywords. case law, legal analysis, agent mentions, named entity recognition, conditional random fields

Introduction

We examine the possibility of automatic detection of agent mentions in case law analysis. This would be an important prerequisite for many applications, such as attribution resolution. It may also become an important component of other applications such as information retrieval or summarization. We assess the hypothesis that a simple sequential model that uses low-level textual features could learn to detect agent mentions automatically (hypothesis 1). Obtaining data for a statistical learning model is expensive. Therefore we explore the relatedness of the task when performed on different areas of law (cyber crime and intellectual property). We first confirm that when a model is trained on decisions from one area and applied to texts from the other domain the performance is lower (hypothesis 2). But we also show that using texts from multiple domains may lead to higher quality predictive models (hypothesis 3).

1. Background and Motivation

Case law analysis is the process of determining which prior court decisions apply to a case, how they apply, and the effect of this application. In the context of judicial decision-

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

making the objective of the analysis could be to generate persuasive case-based arguments. These arguments could play a pivotal role in how a court decides a case. In the American legal system under the common law doctrine of *stare decisis*, like cases are decided alike. [3, p. 9]

Case law analysis encompasses two different, yet closely related, activities. First, a lawyer needs to *identify a set of decisions* that are relevant to argumentation in the given case. Then, from the texts of the decisions one *extracts valuable information* such as: authoritative applications of the rule conditions and concepts to identified situations, a ground truth for testing predictions about outcomes in new cases with new evidence, patterns for successful and unsuccessful argumentation, and guidance in retrieving, extracting, and organizing evidence for new arguments and new situations. [5, p. 176] This is an iterative process where the newly found pieces of information inform search for additional relevant decisions.

Existing legal information retrieval (IR) systems are relatively well suited to support the task of identifying relevant decisions. By means of a search query a lawyer specifies a hypothesis about what words and phrases are likely to occur in relevant decisions. The IR systems are much less equipped to help with the extraction of valuable pieces of information from the texts. Most of the times this needs to be done manually.

It has been extensively argued and shown that computational support for directly retrieving arguments and argument-related information (AR) would be extremely valuable. [4,9] Despite the great promise there is still a considerable gap between the demonstrated automatic analysis capabilities and a full-blown AR system. [2] Due to peculiarities of legal texts even the most foundational natural language processing (NLP) techniques are often performed poorly. One such technique is the detection of agent mentions. Being able to recognize when an agent is mentioned is vital, among many other applications, for attribution resolution. [16] This is why we focus on the capability to detect agent mentions automatically.

2. Task Definition, Proposed Solution, and Working Hypothesis

Detecting agent mentions amounts to recognizing when a word or a phrase denotes an agent. An agent could be any person or organization from informal groups to business companies and governmental entities. As it turns out a typical court decision contains many mentions of agents as shown in the following example:

> The magistrate judge denied the second motion to compel because Mavrix failed to notify the anonymous parties of the pending motion. Mavrix moved the district court for review of the magistrate judge's order, which the district court denied on the basis of the moderators' First Amendment right to anonymous internet speech.

In the short excerpt above there are multiple mentions of a judge, Mavrix (a party), anonymous parties, a court, and moderators—all of these are agents. Since we aim for the maximum possible coverage even words such as possessive adjectives (e.g., his, their) are considered agent mentions.

Apart from recognizing that an agent is being mentioned it would be very useful to understand what kind of agent it is. This is especially true for agents that play specific roles in a case (e.g., a court, a party, or a witness). For this reason we defined a

Agent		
Person		Organization
Attorney	Party	Jury
Judge	Amicus Curiae	Legislator
Expert		Court
Witness		

Table 1. The light-weight 3-level agent types hierarchy. The top-level type Agent is differentiated into the Person and the Organization types. These are further distinguished in the bottom level.

light-weight and easily extensible hierarchy of agents. The hierarchy is schematically depicted in Table 1. Different types of agents are organized into three layers. On the top level there is the Agent type that divides into the Person and Organization types (middle level). These two types are further differentiated into the Judge, Party, Attorney, Witness, Expert, Court, Jury, Amicus Curiae, and Legislator types (bottom level).

The task of detecting agent mentions in texts of the court decisions can be understood in the following way: 1. find all the text spans denoting agents; 2. Classify each such text span with the most appropriate agent type from the hierarchy. We hypothesize that both steps of this task could be performed automatically using a sufficiently well trained sequence labeling model such as conditional random fields (CRF).

We expect that the task depends on the domain, that is the area of legal regulation such as cybercrime or copyright. Intuitively, agent mentions such as "a victim" or "an investigator" are more likely to appear in a cybercrime decision whereas "a copyright holder" would more often appear in a copyright case. We also expect the domains to be related in a sense that some knowledge about detecting the mentions in one domain would be useful in a different one.

3. Related Work

Peters and Wyner [13] underscore the importance of identifying agents in legal documents: "At a more fine-grained level, it is important to access who bears what role with respect to the norm, that is, who is the responsible agent or the receiving party within the action." They employed a combination of pattern-identifying rules, parsing and semantic information about verbs and their arguments as heuristics to identify role bearing agents in European Directives. Similarly, the xmLegesExtractor tool used knowledge-engineered text classification rules and natural language parsing to extract role-playing agents regarding a statutory duty such as addressee, action, and counter-party. [8]

Researchers have also applied supervised machine learning to extract relevant functional elements from multiple states' statutes dealing with public health emergencies including the types of public health system participants who are the acting and receiving agents of regulatory directives. [15] The information is used among other things to construct statutory network diagrams with which to compare the states' regulatory schemes.

Quaresma and Goncalves [14] used parsing for named entity recognition of organizations in a corpus of international agreements from the Euro-Lex site and machine learning to identify types of agreement. The intention was to enrich an ontological index for improving information retrieval.

According to Faiz and Mercer [7] "extraction of many higher order relations is dependent on coreference resolution. ... [A]ugmenting a coreference resolution module in [a] pipeline would be an immediate improvement." For instance, a robust ability to iden-

	# of docs	# of chars	# of tokens	# of sentences	longest	average	shortest
cyber-crime	5	199980	71100	1772	61703 (c)	39996.0 (c)	28306 (c)
					20881 (t)	14220.0 (t)	10414 (t)
					513 (s)	354.4 (s)	250 (s)
intellectual-property	5	247042	90286	2084	75625 (c)	49408.4 (c)	36823 (c)
					27915 (t)	18057.2 (t)	13144 (t)
					729 (s)	416.8 (s)	291 (s)

Table 2. The data set summary statistics. In the last three columns the length is reported in characters (c), tokens (t), and sentences (s).

tify agents referred to in legal decisions is necessary to deal with the problem of attribution, "determining who believes a stated proposition to be true." [17] As Walker argues, "accurate attribution can be a critical task for argumentation mining." For example, it can help to assign legal sentence role types in an annotation pipeline "by distinguishing among ... the testimony of an expert witness, ... or a conclusion or finding of fact by the judge." Automatically detecting distinctions such as between evidence statements and a court's findings of fact could help transform legal IR into argument retrieval. [4]

Some research has focused on identifying agent references in legal decisions. Dozier et al. [6] applied a combination of table lookup, contextual rules, and a statistical model (CRF) to recognize types of entities in captioned legal decisions including jurisdiction, court, and judge. In order to resolve the entities of various types, a SVM model learned to match the extracted entity types and information against authoritative files of actual jurisdictions, courts, and judges.

Al-Kofahi, et al. [1] presented an algorithmic technique that combined parsing, domain knowledge about court hierarchies, and discourse analysis to identify treatment history language in court opinions. Such language includes references to courts as agents as in, "The court in Jones held that ... On the other hand, the district court of Oklahoma, held that ..."

4. Agent Mentions Data Set

We downloaded ten court decisions from the online Court Listener[2] and Google Scholar services.[3] Five of these decisions are from the area of cyber crime (cyber bullying, credit card frauds, possession of electronic child pornography), and five cases involve intellectual property (copyright, trade marks, patents). Detailed information about the texts is provided in Table 2. We use cases from the two different areas of law to measure how well the trained models generalize. We also explore if a model trained for one area of law could improve the performance of a model trained for a different domain.

We created guidelines for manual annotation[4] of the decisions with the types from the hierarchy introduced in Section 2. The two human annotators (the authors) were instructed to aim for the:

1. *Full coverage* – every single word or a phrase that denotes an agent should be annotated with one of the available types.

[2] `www.courtlistener.com`
[3] `scholar.google.com`
[4] Accessible at `luima.org`.

	AGT	PER	ORG	ATT	JDG	EXP	WTN	PTY	AMC	JUR	LEG	CRT
full agreement	.74	.53	.59	.63	.80	.00	.00	.81	.63	.00	.48	.71
partial agreement	.87	.64	.74	.67	.84	.00	.00	.90	.71	.89	.48	.81

Table 3. The inter-annotator agreement for each of the agent mention types showing Agent (AGT), Person (PER), Organization (ORG), Attorney (ATT), Judge (JDG), Expert (EXP), Witness (WTN), Party (PTY), Amicus Curiae (AMC), Jury (JUR), Legislator (LEG), and Court (CRT).

2. *Maximum specificity* – the annotation should be done with the most specific appropriate type (e.g., in case the Agent, Organization, and Legislator types are all appropriate the Legislator type should be used).

For each type the guidelines provide a general definition as well as a couple of examples.[5]

Each decision was annotated by one of the annotators. A small subset (3 decisions) was annotated by both the annotators to measure inter-annotator agreement (see Table 3). We report the full as well as partial agreement. The full agreement is a ratio of the annotations that were created by the both users (i.e., they agree in type and the text span they cover) over all annotations. For partial agreement the annotations are considered to agree if they are of the same type and if they overlap by at least one character.

Table 3 shows that the agreement varies widely across the types. First, it should be noted that the type system is hierarchical. This means that any type also counts as the Agent. When computing the agreement for the Agent type we took into account all the 7004 annotated mentions (not just the 387 where the Agent type itself was marked). Something similar is true of the Person and the Organization types. The .00 agreement for the Expert and the Witness type is due to data sparsity. The agreement was measured on the IP documents. Table 4 shows that these two types were rare on these texts. The .00 full agreement (versus .89 partial agreement) for the Jury type is a systematic error of one of the annotators. The articles ("a", "an", "the") were supposed to be included in the annotations but the annotator failed to do so for the Jury type. As could be expected this error manifests in full agreement but it has no effect on partial agreement.

Table 4 provides detailed statistics of the created annotations. A rather small number of decisions (10) may suggest a relatively small size of the data set. As shown in Table 2 some of the decisions are very long. The total number of annotations (7004) clearly shows that the data set is sufficient for far more than toy experiments. The data set is publicly available.[6]

5. Experiments

5.1. Experimental Designs

We conducted three experiments to test the three hypotheses in this paper. In the *same domain experiment* we assessed the possiblity of detecting the agent mentions (types from Table 1) automatically (hypothesis 1). The goal of this experiment was to determine how well could a sequence labeling model (CRF) separate the signal from the noise for

[5]For example, the definition for the Attorney type is the following: "The Attorney type is reserved for mentions of agents that are known to be attorneys. These usually represent one of the parties or other participants of the proceedings (e.g., amicus curiae)."

[6]Hosted at `luima.org`.

	AGT	PER	ORG	ATT	JDG	EXP	WTN	PTY	AMC	JUR	LEG	CRT
cyber-crime												
# of seq	146	612	236	72	96	14	195	1352	0	82	17	334
# of seq / doc	29.2	122.4	47.2	14.4	19.2	2.8	39.0	270.4	0.0	16.4	3.4	66.8
intellectual-property												
# of seq	241	661	433	76	115	37	34	1668	35	81	16	451
# of seq / doc	48.2	132.2	86.6	15.2	23.0	7.4	6.8	333.6	7.0	16.2	3.2	90.2
total												
# of seq	387	1273	669	148	211	51	229	3020	35	163	33	785
# of seq / doc	38.7	127.3	66.9	14.8	21.1	5.1	22.9	302.0	5.5	16.3	3.3	78.5

Table 4. The summary statistics of the manually annotated agent mentions shows counts for Agent (AGT), Person (PER), Organization (ORG), Attorney (ATT), Judge (JDG), Expert (EXP), Witness (WTN), Party (PTY), Amicus Curiae (AMC), Jury (JUR), Legislator (LEG), and Court (CRT).

the purpose of recognizing the agent mentions. For this experiment the decisions were divided according to the domain from which they came.

In the *different domain experiment* we applied models trained on one area of law to the texts from the other domain. For example, we trained models on a training set of cyber-crime decisions and we evaluated them on on an intellectual property test set. The aim of this experiment was to confirm that the models' performance deteriorates when they are applied to decisions from a different domain (hypothesis 2). If so, it would suggest that the task is domain dependent.

In the *combined domains experiment* we used labeling models trained on one area of law to inform models trained for a different area. For example, predictions of a model trained on the cyber-crime data set were used as features for a model trained on the intellectual property data set. The goal of this experiment was to find out if a model improves when knowledge of another model trained for a different domain is taken into account (hypothesis 3).

In all the three experiments we train a separate CRF model for each agent mention type. Although this is certainly suboptimal, we use the same training strategy and features for all the models. It may be the case that different types (such as the Court or the Attorney) could benefit from a custom-tailored model and contextual features. We reserve fine-tuning of the individual models for future work. 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[7] implementation of CRF. [11,12]

The texts were first tokenized. Each of the tokens is then a data point in a sequence a model operates on and it is represented by a small set of relatively simple low-level textual features. As labels we use the annotation types projected into the BILOU[8] scheme. The features include a token in lowercase, token's signature (a digit maps to "D", lowercase character maps to "c", uppercase to "C"), the token's length, its position within document, whether it is upper case, lowercase, titled, a digit or whitespace. For each token similar features from the three preceding and the three following tokens are included.

[7] `www.chokkan.org/software/crfsuite/`

[8] B: beginning of sequence, I: inside sequence, L: last in sequence, O: outside of sequence, U: unit-length sequence

5.2. Evaluation

To measure performance we use traditional IR metrics—precision (P), recall (R), and F_1-measure (F_1).

$$P = \frac{|Pred \cap Gold|}{|Pred|} \quad R = \frac{|Pred \cap Gold|}{|Gold|} \quad F = \frac{2*P*R}{P+R}$$

Pred is the set of predicted annotations and *Gold* is the set of manually created annotations. In order to determine equality of annotations we used the same two approaches as when computing the inter-annotator agreement—the full (exact) match and the partial (overlap) match.

In the *same domain experiment* we used the leave one out cross-validation on the level of documents. This means that we have conducted the experiment for each of the documents. In a single round one document was a test set and the remaining documents from the same domain were included in the training set. For each type of agent we trained a separate CRF model on the training set. The model was then evaluated on the test set. The point was to see how successful the models are in detecting the agent mentions as compared to the performance of human experts.

For the *different domain experiment* a similar method was used. Again, the experiment was conducted multiple times—once for each document. Instead of using the remaining documents from the same domain as the training set, the documents from the other domain were used. The idea is to compare the performance of these models to the performance of the models trained on the same domains (the preceding experiment).

In the *combined domains experiment* the data from both domains were pooled together. Again, for each document there was a separate round. The point is to compare the performance of these models to that of the models trained on the same domains as well as on the different domains (the two preceding experiments) when applied alone. Our intuition was that at least some knowledge learned in other domain could be transferable.

5.3. Results

Table 5 summarizes the results of the three experiments described in Subsection 5.1. The evaluation metrics are explained in Subsection 5.2. The performance of the models differs considerably across the types but it correlates well across the experiments. That is, if the models trained to detect, say, the Jury type perform well in one of the experiments they perform similarly well in the other two experiments.

Because the type system is hierarchical we took into account all the predicted mentions when computing the metrics for the Agent type (i.e., notwithstanding its type any mention is also an agent). This is also true for the Person and the Organization types. All the other types are at the bottom level. Therefore only those mentions specifically marked with the respective type were considered when assessing the respective models.

The Jury and the Court models are very promising. The Agent, the Organization, the Attorney, the Judge, and the Party models have reasonable performance as well. The performance of the models for the Person and the Legislator types is lower but the models obviously are able to pick some signal. The models for the remaining types perform poorly. In case of the Expert and the Witness types, data sparsity could be the cause.

The models created in the *different domain experiment* tend to have the lowest performance (the middle block of Table 5). This is especially true for the Agent, the Person,

		AGT	PER	ORG	ATT	JDG	EXP	WTN	PTY	AMC	JUR	LEG	CRT
same domain													
exact	P	.74	.65	.79	.67	.47	.00	.56	.73	.17	.87	.50	.81
	R	.36	.17	.39	.25	.16	.00	.04	.36	.03	.56	.06	.69
	F_1	.48	.27	.52	.37	.23	.00	.08	.48	.05	.68	.11	.75
overlap	P	.83	.73	.85	.73	.72	.00	.61	.84	.50	.91	1.0	.87
	R	.40	.19	.42	.27	.24	.00	.05	.41	.09	.59	.12	.74
	F_1	.54	.31	.57	.39	.36	.00	.09	.55	.15	.72	.22	.80
different domain													
exact	P	.67	.48	.70	.59	.46	.00	.00	.63	.00	.85	.27	.80
	R	.28	.09	.39	.18	.20	.00	.00	.23	.00	.63	.09	.68
	F_1	.39	.16	.49	.27	.28	.00	.00	.33	.00	.73	.14	.73
overlap	P	.76	.58	.75	.64	.64	.00	.00	.74	.00	.90	.55	.85
	R	.31	.11	.42	.19	.28	.00	.00	.27	.00	.66	.18	.72
	F_1	.44	.19	.54	.29	.39	.00	.00	.39	.00	.77	.27	.78
combined domains													
exact	P	.70	.66	.73	.68	.52	.00	.52	.69	.22	.88	.45	.79
	R	.37	.23	.43	.35	.26	.00	.06	.34	.06	.69	.15	.72
	F_1	.48	.34	.54	.46	.34	.00	.11	.46	.09	.77	.23	.76
overlap	P	.79	.74	.78	.72	.73	.00	.52	.80	.44	.92	.64	.85
	R	.41	.25	.46	.37	.36	.00	.06	.39	.11	.72	.21	.78
	F_1	.54	.38	.58	.49	.48	.00	.11	.53	.18	.81	.32	.81

Table 5. The performance of the CRF models in automatic detection of agent mentions. The measures used are Precision (P), Recall (R), and F_1-measure (F_1). We assess the models trained to detect Agent (AGT), Person (PER), Organization (ORG), Attorney (ATT), Judge (JDG), Expert (EXP), Witness (WTN), Party (PTY), Amicus Curiae (AMC), Jury (JUR), Legislator (LEG), and Court (CRT).

and the Party types. The best performing models are those created in the *combined domains experiment*. All the models perform at least as well as those that were generated in the *same domain experiment*. The models for the Person, the Attorney, the Judge, and the Jury perform significantly better.

6. Discussion and Future Work

The results summarized in Table 5 clearly show that simple CRF models using low-level textual features are capable of detecting different types of agent mentions automatically. In case of some types (Jury, Court) the performance appears to be sufficient for actual use. In case of some other types (Expert, Witness, Legislator) the performance is clearly too low to produce useful results. For the remaining types it is not clear if the results would have the potential to be useful in practice. This may also depend on the intended application (attribution resolution, summarization).

The performance of the models generated during the *same domain experiment* (top part of Table 5) is better than the performance of the models trained in the *different domain experiment* (middle part of Table 5). This suggests that for each domain there may be certain agent mentions that are rare or non-existent in other domains. In cyber crime one of the prosecuting parties was often mentioned as "the government." This rarely happens in the IP disputes where two private parties are usually involved.

The models created during the *combined domains experiment* (bottom part of Table 5) generally outperformed the models created during both, the *same domain experiment* as well as the *different domain experiment*. This shows that certain patterns in mentioning agents transfer across domains. The Court type mentions appear to transfer very well since even the models trained on the different domain were capable of retaining good performance (e.g., "we" is universally being used to mention the deciding majority).

It is worth emphasizing that the models trained in our experiments are quite simplistic, especially in terms of features they use. While examining the errors it became very clear that simple textual features do not provide sufficient information to detect certain mentions and to distinguish among the different types. One could easily see how using additional resources could lead to dramatic improvements. Take the Amicus Curiae type as an example. The models struggled to distinguish the mentions of this type from mentions of other types, especially the Party and the Organization type. Yet the amici are almost always listed in the header of the decision in a manner that could often allow detection through simple regular expression matching. It is quite likely that detection of the amici in the header and using the detected tokens as contextual features could raise the performance of our models from very bad to excellent.

There are multiple aspects of this work that we would like to address (or see addressed) in future. For some of the mention types (Expert, Witness) we encountered the data sparsity problem. This issue could be affecting other types, too, even though it does not manifest that clearly. It would make sense to enrich the data set with additional documents (perhaps from other areas of law). An interesting option would be to include annotated documents from courts outside the U.S. (e.g., the EU's Court of Justice).

We have defined the limited type hierarchy that includes only the most basic types of agents that are regularly mentioned in decisions (see Table 1). These are by no means all the types that would be of interest for automatic detection. Some of the types that are already included could be further differentiated into subcategories (e.g., Party to Plaintiff, Defendant, Apellant). Thus extending the type hierarchy and annotating the corpus with the new (extended) types would be another way to continue in this work.

The models that we used are fairly simple, especially in terms of the low-level textual features they operate on. Above we have discussed how using more advanced features could lead to considerable improvements. Although, CRF is a decent model for this task some more recent sequence labeling models (e.g., long short-term memory networks) are likely to perform even better provided there is enough data to train them.

Assuming we are able to detect the agent mentions with sufficient accuracy, coreference resolution is a traditional task in natural language processing. The goal in coreference resolution is to determine which words or phrases refer to the same object. In the context of agent mentions this would mean finding out which mentions denote the same agent (e.g., mentions such as "we", "our", "the majority", "this court" could all denote the same agent in a decision).

The ultimate goal is to apply this work in practice. One such application could be automatic attribution resolution. It would be of immense value for a system to determine if a certain interpretation of a legal rule is advanced by the deciding majority, a dissenting judge, or one of the parties. Successful attribution resolution would greatly improve legal IR, argumentation mining, or automatic summarization of legal documents.

7. Conclusions

In this paper we examined the possibility of automatically detecting agent mentions in case law analysis. We have shown that: (i) with varying degree of accuracy it is possible to detect the mentions of different agent types automatically; (ii) the task is domain dependent in a sense that prediction models trained on one area of law do not perform as

well for a different area; and (iii) there is relatedness between domains allowing the use of data from one area of law to improve performance of a model intended for another area. It is our hope that this work will stimulate further research in detecting agent mentions in legal texts. For this reason we release the data set that was created to facilitate the experiments described in this paper. We leave plenty of space for further improvements.

Aknowledgements

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] Al-Kofahi, Khalid, Brian Grom, and Peter Jackson. "Anaphora resolution in the extraction of treatment history language from court opinions by partial parsing." *Proceedings of the 7th international conference on Artificial intelligence and law*. ACM, 1999.

[2] Ashley, Kevin D. *Artificial Intelligence and Legal Analytics*. Cambridge University Press, 2017.

[3] Ashley, Kevin D. *Modeling legal arguments: Reasoning with cases and hypotheticals*. MIT press, 1991.

[4] Ashley, Kevin D., and Vern R. Walker. "From Information Retrieval (IR) to Argument Retrieval (AR) for Legal Cases: Report on a Baseline Study." *JURIX*. 2013.

[5] Ashley, Kevin D., and Vern R. Walker. "Toward constructing evidence-based legal arguments using legal decision documents and machine learning." *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law*. ACM, 2013.

[6] Dozier, C., Kondadadi, R., Light, M., Vachher, A., Veeramachaneni, S., and Wudali, R. "Named entity recognition and resolution in legal text." Semantic Processing of Legal Texts. Springer Berlin Heidelberg, 2010. 27–43.

[7] Faiz, Syeed Ibn, and Robert Mercer. "Extracting higher order relations from biomedical text." *Proceedings of the First Workshop on Argumentation Mining*. 2014.

[8] Francesconi, Enrico. "An Approach to Legal Rules Modelling and Automatic Learning." *JURIX*. 2009.

[9] Grabmair, Matthias, Ashley, K. D., Chen, R., Sureshkumar, P., Wang, C., Nyberg, E., and Walker, V. R. "Introducing LUIMA: an experiment in legal conceptual retrieval of vaccine injury decisions using a UIMA type system and tools." *Proceedings of the 15th International Conference on Artificial Intelligence and Law*. ACM, 2015.

[10] Grabmair, M., Ashley, K. D., Hwa, R., and Sweeney, P. M. "Toward Extracting Information from Public Health Statutes using Text Classification Machine Learning." *JURIX*. 2011.

[11] John Lafferty, Andrew McCallum, Fernando Pereira, and others. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning*, ICML, Vol. 1. 282–289.

[12] Naoaki Okazaki. 2007. CRFsuite: a fast implementation of Conditional Random Fields. (2007).

[13] Peters, Wim, and Adam Z. Wyner. "Legal Text Interpretation: Identifying Hohfeldian Relations from Text." *LREC*. 2016.

[14] Quaresma, Paulo, and Teresa Goncalves. "Using linguistic information and machine learning techniques to identify entities from juridical documents." *Semantic Processing of Legal Texts*. Springer Berlin Heidelberg, 2010. 44–59.

[15] Savelka, Jaromir, Matthias Grabmair, and Kevin D. Ashley. "Mining Information from Statutory Texts in Multi-Jurisdictional Settings." *JURIX*. 2014.

[16] Walker, Vern R. "The Need for Annotated Corpora from Legal Documents, and for (Human) Protocols for Creating Them: The Attribution Problem." *Dagstuhl Seminar on Natural Language Argumentation: Mining, Processing, and Reasoning over Textual Arguments*, 2016.

[17] Walker, Vern R., Parisa Bagheri, and Andrew J. Lauria. "Argumentation Mining from Judicial Decisions: The Attribution Problem and the Need for Legal Discourse Models." 2015.