ECHR: Legal Corpus for Argument Mining

Prakash Poudyal[1], Jaromír Šavelka[2], Aagje Ieven[3], Marie Francine Moens[4],
Teresa Gonçalves[5], Paulo Quaresma[6]

Department of Computer Science & Engineering, Kathmandu University, Nepal[1]
School of Computer Science, Carnegie Mellon University, USA [2]
Katholieke Universiteit Leuven, Faculty of Law, Leuven, Belgium [3],[4]
Department of Informatics, University of Evora, Portugal[5],[6]

prakash@ku.edu.np[1], jsavelka@andrew.cmu.edu[2], aagje.ieven@gmail.com[3],
sien.moens@cs.kuleuven.be[4], tcg@uevora.pt[5], pq@uevora.pt[6]

Abstract

In this paper, we publicly release an annotated corpus of 42 decisions of the European Court of Human Rights (ECHR). The corpus is annotated in terms of three types of clauses useful in argument mining: premise, conclusion, and non-argument parts of the text. Furthermore, relationships among the premises and conclusions are mapped. We present baselines for three tasks that lead from unstructured texts to structured arguments. The tasks are argument clause recognition, clause relation prediction, and premise/conclusion recognition. Despite a straightforward application of the bidirectional encoders from Transformers (BERT), we obtained very promising results (F_1 0.765 on argument recognition, 0.511 on relation prediction, and 0.859/0.628 on premise/conclusion recognition). The results suggest the usefulness of pre-trained language models based on deep neural network architectures in argument mining. Because of the simplicity of the baselines, there is ample space for improvement in future work based on the released corpus.

Keyword: European Court of Human Rights (ECHR), argumentation, argument mining, bidirectional encoders from transformers (BERT)

1   Introduction

Texts can be categorized into subjects such as law, philosophy, computing, science, etc., and annotated with information appropriate for the purpose of the research (e.g. argument mining, named entity recognition, etc.). This information is collectively called a corpus and forms a critical component of this research. The corpora are a source of knowledge for creating certain rules and regulations (i.e. a model) which are used in statistical and hypothetical tests. Although influenced by many other factors, the performance of statistical approaches still depends primarily on the quality and size of the corpus. Therefore during the process of creating a corpus, annotators need to prioritize their ability to maintain quality and quantity via inter-annotator agreement (Artstein, 2017). While creating corpora, it is desirable to ensure that a system will achieve the highest possible accuracy. Nevertheless, creating and constructing corpora is labor-intensive, as it is complex, time-consuming and requires experts who need to be well versed in the corresponding corpus field. For example, in the case of legal corpora, the annotating experts must be lawyers and should also be familiar with the legal arguments. With these constraints, there are a limited number of corpora available in the respective fields.

In this paper a corpus of ECHR decisions is released to the public with a well-structured electronic format[1]. The corpus is annotated with a compact argumentation type system (3 types). Pointers to the work where the corpus has been already used prior to its release here are provided. The annotation process and the tools used are explained in detail. The resulting corpus is described and several tasks where it can be employed are proposed.

The rest of the paper is organized as follows: Section 2 presents the related work focusing on various corpora that are publicly available. Section 3 introduces the ECHR's Case Law. Section 4 describes the annotation types, the annotation process, and the tools that were used for annotation. Section 5 presents the summary statistics related to the resulting corpus. Section 6 presents the baselines for three fundamental argument mining tasks.

2 Related Work

A corpus is typically a representation of text, language, image, video, audio, and subject that is annotated with certain signs for a specific purpose. The performance of a predictive system based upon a corpus also depends on its quality and size. Depending on the goal, corpora are collected from different domains, such as Newspaper, Legal, Political, Scientific and Persuasive essays and also in various languages like English, Portuguese, German, Greek.

Lippi and Torroni (2016) mention that annotating corpora is complex, expensive and requires experts who are well versed in the corresponding field to ensure that annotations are correct. Further, the paper also describes the corpora of various domains concerning the argument mining area. Palau and Ieven (2009) deal with the theoretical aspects of the structure present in legal corpora; they highlight the different critical points humans need to encounter when applying theory to real argumentation and emphasize the association between real arguments and the theories that describe those arguments.

Several research centers in the world are devoted to developing corpora. One such center that plays an important role in many aspects of argumentation, from theoretical to practical, is Arg-tech Centre[2]. Its one of the goals is to develop freely available software tools to aid the researcher in the argumentation field. There are more than 50 corpora from different sectors offered in AIFdb (Bex et al., 2013). These corpora are available in different formats, such as SVG, PNG, DOT, JSON, LKIF, RTNL, RDF, PL. A popular corpus named Araucaria (Reed et al., 2008) was developed in the Arg-tech Centre. The granularity of the corpus was 'claim' and 'premise'. The corpus was collected from 19 newspapers (from the UK, US, India, Australia, South Africa, Germany, China, Russia and Israel), 4 parliamentary records (in the UK, US and India), 5 court reports (from the UK, US and Canada), 6 magazines (UK, US and India), and 14 other online discussion boards and *cause* sources such as the Human Rights Watch (HuRW) and the GlobalWarming.org.

Another research lab, Ubiquitous Knowledge Processing (UKP) Lab [3], TU Darmstadt, Germany, is dedicated to developing natural language and machine learning tools and techniques. One of their activities is to develop corpora for argument mining in English and German. Habernal and Gurevych (2017) created the *Argument Annotated User-Generated Web Discourse* consisting of 90,000 tokens from 340 documents with the datasets prepared using Toulmin argument classification (Backing, Claim, Premise, Rebuttal, Refutation). Habernal and Gurevych also created the *UKPConvArg1 Corpus*, a user-generated Web content corpus that consists of 11,650 argument pairs (Habernal and Gurevych, 2016b) and a new crowd-sourced benchmark data-set that contains 9,111 argument pairs labeled with 17 categories (Habernal and Gurevych, 2016a).

Kwon et al., (2006) developed the corpus based on public comments on the Environmental Protection Agency (EPA) standards rules on hazardous pollutants. Two annotators were involved in categorizing 630 documents in terms of 9 categories achieving an agreement of 0.60 on Cohen's Kappa coefficient. Rosenthal and McKeown (2012) created a corpus from two datasets: 285 LiveJournal[4] blog spots and 51

---

[1] http://www.di.uevora.pt/~pq/echr/
[2] http://www.arg-tech.org/
[3] https://www.ukp.tu-darmstadt.de/ukp-home/
[4] https://www.livejournal.com/

Wikipedia discussion forums; the datasets were annotated for the purpose of identifying claims (the ratio of claim vs. not claim is 60:40 in LiveJournal and 64:36 in Wikipedia). Aharoni et al., (2014) present an argumentative structure dataset consisting of 33 controversial topics; the corpus is derived from 586 Wikipedia articles and the authors state that the corpus was constructed (manual annotation) with great attention to detail.

Goudas et al., (2014) annotated 204 documents in Greek related to renewable energy. The documents contain 16,000 sentences and were collected from social media, news, blogs, and microblogs. 760 sentences were annotated with premise and claim at the clause-level. Similarly, Sardianos et al., (2015) choose 300 news items (sports, politics, economics, and culture, etc.) in the Greek language to annotate; two post-graduate students were assigned to annotate the corpus and since they were only moderately experienced, guidelines were provided to describe the identification of arguments focusing on discourse markers such as *because*, *in order to* and *but*. Each annotator was assigned to annotate 150 documents with argument components, and the final version of the corpus contained 1191 argument components.

3   ECHR's Case Law

Case-law documents are written using detailed information from the stakeholders of the court, factual information from the defendant, allegations made by the plaintiff, arguments from both parties, and a decision made by the judge. After collecting all information, it needs to be structured to be useful. To structure the data, and also to know the location of the components of the arguments, it is necessary to analyze and determine the content of each section available via case-law. From the website[5], it is known that case-law is divided into seven categories: Judgment, Decision, Communicated Case, Legal Summary, Advisory Opinion, Report and Resolution. Of these seven categories, two categories (Judgment and Decision) are found in the ECHR Corpus.

In the ECHR Corpus, there are 20 Decisions and 22 Judgments[6] issued by a chamber of seven judges ruling on the admissibility and merits of the cases. Both categories represent similar information, however, the 'Decision category' presents the information briefly (the average word length is 3500 words) in the corpus whereas, in the case of Judgments, more detailed information is available (an average word length of 10000 words). The Decision case-law documents are divided into six sections: i) Introduction, ii) The Facts, iii) Complaints, iv) Proceedings before the Commission, v) The Laws and vi) For the Reason. Judgment case-law documents are divided into eight sections: i) Introduction, ii) Procedure, iii) As the Facts, iv) The Circumstances of the Case, v) Proceedings before the Commission, vi) Final Submissions to the Court, vii) As to the Law, and viii) For the Reason.

The case-law documents begin with introducing the stakeholders of the courts (President, Judge, Registrar and Deputy Registrar, Lawyers, Plaintiff, Defendants, and their agents), with their designations, plaintiff, defendant and other members involved in the case. After this, procedure and facts regarding the plaintiff are described. The facts describe an overview of the case that includes information from previous cases, the reason for making an allegation and the chronological sequence of events. The structure of the case-law varies depending upon the exact laws. This information is included as necessary, meaning that the case will vary in length depending upon the case-law and any other essential information that is included. After providing facts of the plaintiff and defendant, the case-law includes the discussions held in the court based upon the allegation made by the plaintiff are presented. Likewise, the defendant provides the reason and claim from their perspective and returns a response to the claim made by the plaintiff. After several discussions and arguments presented by both parties, the Judge renders his decision. Detail information about ECHR case law documents can be found in (Mochales and Moens, 2008).

4   Annotation

The Language Intelligence and Information Retrieval Research Lab at the KU Leuven hired two lawyers to annotate the ECHR case-law documents (Palau and Ieven, 2009) . The annotators were given an

---

[5]https://hudoc.echr.coe.int/eng
[6]Released before 20 October 1999 by the European Commission on Human Rights.

argumentation scheme formalism and guidelines that describe the arguments.

Once the annotation was completed, they were compared and the Kappa inter-rater agreement tally was found to be 0.58. A third lawyer was selected to analyze the annotations and found that the main reason for the discrepancies was due to a different demarcation of argument boundaries or, put another way, to the ambiguity that is found in argumentative structure. Subsequently, a fourth annotator was selected and was given new guidelines, new sets of comments and recommendations. His annotation achieved 80% agreement which is quite a significant gain. Additional information regarding the evaluation of the corpus annotation process can be found in (Palau and Ieven, 2009), (Mochales and Moens, 2008).

The corpus is composed of 42 decisions. There are 1951 premises and 743 conclusions (note some argument constituents are premises/conclusions for more than one argument, and some constituents are both premises of one argument and conclusion for another).

- Premise – one or more premises are bound to exactly one conclusion.

- Conclusion

- Non-argument

The documents were annotated in the Gloss annotation environment developed at the University of Pittsburgh. Gloss is a lightweight tool focused on semantic annotation of textual documents. Through a small number of individual components it supports the whole annotation process, including corpus assembly, type system definition, document annotation, as well as quality control. The system is equipped with simple identity and role management that facilitates basic security as well as the ability of multiple users performing various roles within a project. Gloss was already used in multiple research projects, e.g. in Savelka and Ashley (2018; 2017).

The first step typically involves uploading the plain text documents; these can be organized in one or more collections. Next, the type system needs to be created. Gloss is very flexible with respect to type systems. It offers three built-in types (Annotation, Object, and ValueSet) that function as building blocks from which a type system of arbitrary complexity may be assembled. Types may be defined with an arbitrary number of attributes. Inheritance and composition of types are supported as well. Finally, a task may be created using one or more document collections and one or more type systems. Multiple users may be involved in a task either as annotators or editors (quality control).
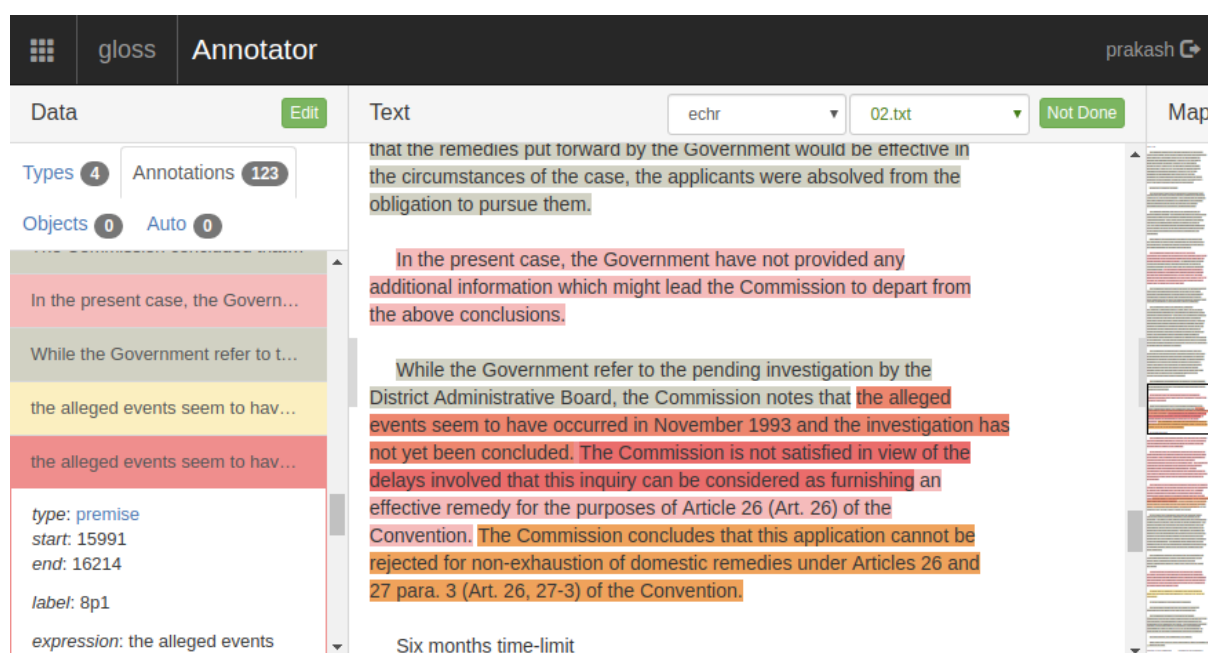

Figure 1: Annotation interface (Gloss) used in this work.

For the annotation itself, a user is presented with a list of documents from which he can select the one to work on. The available types are listed in the left pane. As the user creates annotations, an interactive list is being populated. The annotations may be easily edited or deleted after being created. The annotation is created through highlighting the respective span of text and picking the respective type from the pop-up menu. The annotation user interface is shown in Figure 1.

5 Resulting Corpus

Summary statistics about the resulting corpus are reported in Table 1; the data structure of the published corpus is sketched in Figure 2.

	Premises	Conclusions
Minimum	8	4
Mean	47.74	18.29
Maximum	147	50
Total	1951	743

Table 1: Summary dataset statistics at document level.

```json
[
    {
        "name": "Case Name",
        "text": "Lorem ipsum dolor sit amet, consectetur adipiscing elit.",
        "clauses": [
            {
                "id": "5d3878c43e582511aa1cbdee",
                "type": "non-argument/argument",
                "start": 5,
                "end": 15
            }
            ...
        ],
        "arguments": [
            {
                "premises": [
                    "5d3878c43e582511aa1cbdee", ...
                ],
                "conclusion": "5d3878c43e582511aa1cbdee"
            }
            ...
        ]
    }
    ...
]
```

Figure 2: The data structure for storing the dataset.

The published JSON file is a list of the 42 cases. Each case is an object with the following fields:

- *name* – Stores the name of the file

- *text* – Stores the full-text of the case

- *clauses* – Lists the clauses annotated within the corresponding case

- *arguments* – Lists the argument structure of the clause

Each clause is an object with the following fields:

- *id* – Unique identifier

71

- *type* – A binary indicator of argument membership.

- *start* – Character offset where the clause starts in the *text*

- *end* – Character offset where the clause ends in the *text*

Each argument is an object with the following fields:

- *premises* – Lists the unique identifiers (*id*) of the clauses that are the premises of the argument

- *conclusion* – The unique identifier (*id*) of the clause which is the conclusion of the argument

Note that the information about the clause type is not provided directly in the clause objects. This decision is due to the fact that a clause can be of multiple types via its membership in different arguments. Therefore, the types of the clauses are encoded in the argument structures themselves: a clause which is not a part of any argument structure is of the *Non-argument* type; a clause which is listed as a premise in one or more argument structures is of the *Premise* type; the clause which is a conclusion of any argument structure is of the *Conclusion* type.

6 Baselines

In this section we present the baselines on the three tasks defined in the past that lead from unstructured texts to structured arguments. The tasks are argument clause recognition, clause relation prediction, and premise/conclusion recognition. As a basis for the classification algorithms we employ the RoBERTa pre-trained language model (Liu et al., 2019) which is a variation on the original BERT model (Devlin et al., 2018) with improved pre-training phase. A small layer is placed on top of the model to handle the classification. We work with the base version of RoBERTa to make sure the baselines are simple and easily reproducible. The base RoBERTa model runs perfectly fine on the GPUs one can access for free on Google Colaboratory.[7]

In all the experiments 5-fold cross-validation was used. The models were fine-tuned on each of the specific tasks for 15 epochs for each of the folds. In each of the iterations 20% of the data set was used for testing. Of the remaining 80% documents 20% were used for validation. After the end of each epoch we saved the current version of the model and recorded its performance on the validation set. For the evaluation on the test set we used the version of the model with the best F_1 score as evaluated on the validation set.

6.1 Argument Clause Recognition

In this task, the goal is to predict if a clause belongs to an argument or not. Specifically, this means recognizing if the clause is annotated with any of the two argument types, i.e. the premise or the conclusion, or with the non-argument type. The prior work understood this task as a binary classification problem and employed traditional ML methods, such as Random Forest or Support Vector Machines (SVM). Poudyal et al. (2016) achieved the best results ($F_1 = 0.705$) with the system based on SVM using a combination of low-level linguistic (word n-gram, POS n-grams) and hand-crafted features (e.g., the section of the document the clause comes from). Mochales and Moens (2011) use a Maximum Entropy classifier and report accuracy of 80%.

Here, we understand the task as binary classification as well. As a basis for the classification algorithms we employ the RoBERTa pre-trained language model as explained above. A small layer is placed on top of the model to handle the classification. Unlike in the case of SVM or Random Forest, one cannot provide the model with hand-crafted features. However, it is an apparent feature of this data set that an overwhelming majority (over 99%) of argument clauses is present in "AS TO THE LAW"/"THE LAW" section of the case texts. Since the ECHR cases have a uniform structure and the section can be reliably detected using text matching we leveraged the heuristic. We automatically predict every single clause outside of "AS TO THE LAW"/"THE LAW" section as being non-argument. The model is then trained

---

[7] https://colab.research.google.com/

Task	Precision	Recall	F-measure
Argument clause recognition	0.697	0.848	0.765
Argument relations mining	0.502	0.521	0.511
Premise recognition	0.832	0.887	0.859
Conclusion recognition	0.589	0.672	0.628

Table 2: The performance of the simple baseline systems modeling the three fundamental steps in argumentation mining as understood in this work.

on the clauses coming from "AS TO THE LAW"/"THE LAW" section only. In the evaluation phase, we simply considered the argument clauses outside of the section as false negatives.

The results of the experiment are reported in Table 2 (first row). The F_1 score on the positive class of 0.765 appears promising. However, we would like to emphasize that it is not our primary goal to achieve maximum performance. The main purpose of the system is to provide a straightforward, yet competitive, baseline. It is our assumption that significantly outperforming this baseline would require a system that would become a valuable contribution to the field of argument mining, especially argument mining from legal texts. Second, maximum caution is required when comparing to prior work in this case. Some small alterations of the corpus might have occurred (e.g., correction of obvious mistakes). Hence, the comparison is almost certainly not 1-to-1.

6.2 Argument Relation Mining

This task assumes that we have successfully identified the argument clauses. The ultimate goal is to assign the clauses into groups (i.e., arguments). The situation is somewhat complicated by the fact that a single clause may be a member of multiple arguments, e.g., it is a premise in one argument and conclusion in the other. Poudyal et al. (2019) use the Fuzzy c-means clustering algorithm (Bezdek et al., 1984) to perform this task. They report a macro F_1 (on a document level) of 0.50 and cluster purity of 0.50. Mochales and Moens (2011) approached the task differently. They used a manually created context-free grammar to produce full argument parses. They report an accuracy of around 60%.

As shown by the prior work this problem is very difficult. It is probably the crucial bottleneck in mining arguments as understood in this work. Hence, we propose a baseline that transforms the problem into a simpler one with very well defined and intuitive evaluation metrics. We understand the task as a sentence pair classification. Specifically, given a pair of argument clauses coming from the same document we want to predict if they are members of the same argument. Thus, we effectively cast the task into a binary classification while allowing for the membership of individual clauses in multiple arguments. Of course, the drawback of this approach is that one does not receive a final grouping. An additional phase would be required to turn the output into the arguments. This is left for future work.

As a basis for the classification algorithms we again employ the RoBERTa model with a small classification layer on top of it. Note that using every possible pairing of argument clauses retrieved from a document would not be a sound strategy. This would lead to a highly unbalanced data set where only a very small number of examples would be positive. Instead one can leverage the fact that argument clauses typically cluster in small proximity to each other. For training we only consider the pairs of arguments that are no more than five sentences apart. The window of five was set heuristically as it appeared neither too small nor too large. Any too argument clauses that are further apart than the set threshold are automatically predicted as not being in the argument relation. The actual relations that exist but are missed using this strategy are considered as false negatives for the purpose of the evaluation.

The results of the experiment are reported in Table 2 (middle row). They confirm what has been shown earlier, i.e., this part of the task is very challenging.

6.3 Premise/Conclusion Recognition

This task also assumes that we have successfully identified the argument clauses. The goal is to decide which of the clauses are conclusions and which of them are premises. Note that it is possible for a single clause to be a premise in one argument and a conclusion in another one. Mochales and Moens (2011) appear to train two SVM models, one to recognize premises and one to recognize conclusions. Notably, they appear to use highly sophisticated hand-crafted features, e.g., a category of the preceding sentence, discursive cues, or type of main verb. They report F_1 score of 0.681 for premises and 0.741 for conclusions.

We follow the example set in Mochales and Moens (2011) and approach this task by training two separate binary classifiers, one for each clause type. An alternative would be to understand this task as a multi-label classification. For each classifier the task is to decide if the clause is a premise/conclusion or not. As a basis for the classification algorithms we again employ the pre-trained RoBERTa model with a small classification layer on top of it.

The results of the experiment are reported in Table 2 (third and fourth row). Interestingly, in Mochales and Moens (2011) it was reported that recognizing premises is more difficult than conclusions. Our experiments show the exact opposite. This is most certainly due to the fact that the hand-crafted features used in Mochales and Moens (2011) were much more suitable for recognizing conclusions than premises.

7 Conclusions

In this paper, the ECHR Corpus is described in detail. The procedure undertaken for annotating the components of the arguments in the case-law is described, and an analysis of the quality of the corpus, including its statistical nature and structure is offered.

It is a characteristic of legal corpora that many, if not all, case law documents are interconnected to each other through citations. For that, as a future work it is necessary to annotate the related content/information provided by the citation link. For example, in the ECHR corpus excerpt

> "**The notion of security of person has not been given an independent interpretation** *(see in this respect Selçuk and Asker v. Turkey, nos. 23184/94 and 23185/94, Commission's report of 28 November 1996, §§ 185-187)."*

the first phrase in bold is the conclusion, and the text in italics is the premise. In future work, the annotator would check the report mentioned and annotate that sentence as a premise. This work might help in achieving higher classification performances.

Acknowledgements

References

Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. 2014. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *ArgMining@ACL*, pages 64–68.

Ron Artstein, 2017. *Inter-annotator Agreement*, pages 297–313. Springer Netherlands, Dordrecht.

Floris Bex, John Lawrence, Mark Snaith, and Chris Reed. 2013. Implementing the argument web. *Communications of the ACM*, 56(10):66–73.

James C. Bezdek, Robert Ehrlich, and William Full. 1984. Fcm: The fuzzy c-means clustering algorithm. *Computers and Geosciences*, 10(2):191 – 203.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Theodosis Goudas, Christos Louizos, Georgios Petasis, and Vangelis Karkaletsis. 2014. Argument extraction from news, blogs, and social media. In *Hellenic Conference on Artificial Intelligence*, pages 287–299. Springer.

Ivan Habernal and Iryna Gurevych. 2016a. What makes a convincing argument? empirical analysis and detecting attributes of convincingness in web argumentation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1214–1223.

Ivan Habernal and Iryna Gurevych. 2016b. Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional lstm. In *ACL (1)*.

Ivan Habernal and Iryna Gurevych. 2017. Argumentation mining in user-generated web discourse. *Computational Linguistics*.

Namhee Kwon, Stuart W Shulman, and Eduard Hovy. 2006. Multidimensional text analysis for erulemaking. In *Proceedings of the 2006 international conference on Digital government research*, pages 157–166.

Marco Lippi and Paolo Torroni. 2016. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology (TOIT)*, 16(2):10.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Raquel Mochales and Marie-Francine Moens. 2008. Study on the structure of argumentation in case law. In *Proceedings of the 2008 Conference on Legal Knowledge and Information Systems*, pages 11–20.

Raquel Mochales and Marie-Francine Moens. 2011. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22.

Raquel Mochales Palau and Aagje Ieven. 2009. Creating an argumentation corpus: do theories apply to real arguments?: a case study on the legal argumentation of the echr. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 21–30. ACM.

Prakash Poudyal, Teresa Goncalves, and Paulo Quaresma. 2016. Experiments on identification of argumentative sentences. In *Proceeding of 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA) Chengdu, China*, Dec.

Prakash Poudyal, Teresa Gonçalves, and Paulo Quaresma. 2019. Using clustering techniques to identify arguments in legal documents. In *ASAIL@ ICAIL*.

Chris Reed, Raquel Mochales Palau, Glenn Rowe, and Marie-Francine Moens. 2008. Language resources for studying argument. In *Proceedings of the 6th conference on language resources and evaluation-LREC 2008*, pages 91–100.

Sara Rosenthal and Kathleen McKeown. 2012. Detecting opinionated claims in online discussions. In *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*, pages 30–37. IEEE.

Christos Sardianos, Ioannis Manousos Katakis, Georgios Petasis, and Vangelis Karkaletsis. 2015. Argument extraction from news. In *ArgMining@ HLT-NAACL*, pages 56–66.

Jaromír Savelka and Kevin D Ashley. 2017. Detecting agent mentions in us court decisions. In *JURIX*, pages 39–48.

Jaromir Savelka and Kevin D Ashley. 2018. Segmenting us court decisions into functional and issue specific parts. In *JURIX*, pages 111–120.