Towards Annotating and Extracting Textual Legal Case Elements

Adam Wyner
University of Leeds

Abstract.
In common law contexts, legal cases are decided with respect to precedents rather than legislation as in civil law contexts. Legal professionals must find, analyse, and reason with and about cases drawn from a set of cases (a case base). A range of particular textual elements of a case may be relevant to query and extract. Commercial providers of legal information allow legal professionals to search a case base by keywords and meta data. However, the case base and the search tools are proprietary, of limited, non-extensible functionality, and are restricted access. Moreover, no provider applies natural language processing techniques to the cases for text analysis, XML annotation, or information acquisition. In this paper, we discuss an initial experiment in developing and applying natural language processing tools to cases to produce annotated text which can then support information extraction.

Keywords: Text Analysis, Legal Cases, Ontologies

1. Introduction

In common law contexts, judges and juries decide a legal case to follow previously decided cases (precedents) rather than legislation as in civil law contexts.[1] The set of such cases is the legal case base. Legal professionals must find, analyse, and reason with and about cases drawn from the case base in the course of arguing for a decision in a current undecided case. A range of elements of cases may be relevant to query and extract such as the citation index, participants, locale, jurisdiction, representatives, judge, prototypical fact patterns (factors), applicable law, and others. Commercial providers of legal information allow legal professionals to search the case base by keywords and meta data. However, the case base and search tools are proprietary, of limited, non-extensible functionality, and are restricted access. Moreover, no provider works with Semantic Web functionalities such as ontologies or rich XML annotations, nor are natural language processing techniques applied to the cases to support analysis to acquire information.

Text annotation of unstructured linguistic information is a significant, difficult aspect of the "knowledge bottleneck" in legal information processing. In this paper, we apply natural language processing tools to textual elements in cases, which are unstructured text, to produce annotated text, from which information can be extracted, thus contributing to overcoming the bottleneck. The extracted information can then be submitted to further processes.

[1] Correspondence to Adam Wyner adam@wyner.info.

Where the annotations are associated with an ontology (Wyner and Hoekstra, 2010) along with an associated case based reasoner (Wyner and Bench-Capon, 2007), then we make progress towards a textual case based reasoning system which enables processing from natural language case decisions in the case base to generated decisions in novel cases (Weber et al, 2005a). However, this paper focuses on the initial development in annotating cases with respect to case elements.

The paper is a feasibility study for future research on information extraction of case elements. [2] In this paper, we focus on case elements rather than case factors (see (Wyner and Peters, 2010)).

In 2, we discuss background and materials. In 3, we present the methodology, which uses the General Architecture for Text Engineering(GATE) system, sample components of system, sample results, and a work flow for further refinement.[3] Finally, in 4, we review the paper and outline future work to evaluate and improve our results.

2. Background and materials

Legal case based reasoning with factors has been a topic of central concern in artificial intelligence and law. For our purposes, there are two main branches of research. One branch, knowledge representation and reasoning systems, requires a knowledge base that is constructed by manual analysis (cf. (Hafner, 1987), (Ashely, 1990), (Rissland et al, 1996), (Aleven, 1997), (Wyner and Bench-Capon, 2007)). However, this branch of research does not address the knowledge bottleneck, which is the extraction of information to compose the knowledge base.

The other branch, information extraction, addresses the bottleneck using natural language processing techniques which identify informative components of the text and annotate them with XML. The annotated information can be extracted with *XQuery*. Thus, the content of the documents can be identified from its source linguistic realisation. There are a range of areas where information extraction of legal texts has been carried out: ontology construction ((Lame, 2004) and (Peters, 2009)), text summarisation ((Moens et al, 1997) and (Hachey and Grover, 2006)), extraction of precedent links (Jackson et al, 2003), and factor analysis ((Ashley and Brüninghaus, 2009) and (Wyner and Peters, 2010)). We focus on information extraction of case elements, which contributes to this previous work.

The branches are related since the extracted information can be represented in some knowledge base and reasoned with. For case based reasoning with factors as in (Aleven, 1997), we extract factors; for reasoning about

[2] Contact the author for materials.

[3] For GATE, see http://gate.ac.uk/.

precedential relations among cases (overturned, affirmed, and so on), we extract citation indices and relational terms. As legal cases are not just about the law *per se*, but about some content area (e.g. intellectual property, family law, etc) and human properties and artifacts (e.g. instruments and property), one might suppose that all of human knowledge and experience is potentially under the scope of the law and so potentially to be extracted, put in a knowledge base, and reasoned with (cf. works on legal knowledge representation (Peters et al, 2007), (Scheighofer and Liebwald, 2007), (Hoekstra et al, 2009), and (Gangemi et al, 2005)). Yet, (Wyner and Hoekstra, 2010) argue that the focus should be on information which has a legal definition or function, leaving aside high level, non-legal domain information (e.g. events/processes, causation, time, and so on).

In this light and in the current paper, we are interested in case information that would be relevant to searching for or extracting information from cases. For reasons of space, we only give a sample of the information we searched for and annotated:

- Case citation, cases cited, precedential relationships.
- Names of parties, judges, attorneys, court sort....
- Roles of parties, meaning plaintiff or defendant, and attorneys, meaning the side they represent.
- Final decision.

With respect to these features, one would want to make a range of queries (using some appropriate query language) such as:

- In what cases has company X been a defendant?
- In what cases has attorney Y worked for company X, where X was a defendant?

As we initially based our work on information extraction from California Criminal Courts in (Bransford-Koons, 2005), developing and modifying lists and rules, we worked with a legal case base of cases from the United States. (Bransford-Koons, 2005) reports working with 47 criminal cases drawn from the California Supreme Court and State Court of Appeals. However, only two cases are given as samples and for which we have access; for this feasibility study, we give examples from these cases. (Bransford-Koons, 2005) uses GATE (described below) and OPENCYC, which is a repository of common sense rules. We do not consider OPENCYC here. To show the feasibility of the approach, we provide preliminary results on this very small corpus of *People v. Coleman 117 Cal.App.2d 565* and *In re James M., 9 Cal.3d 517*.

3. Methodology using GATE

We use the GATE framework (Cunningham et al, 2002). GATE Developer is an open source desktop application written in JAVA and for linguists and text engineers. Using a GUI, it allows a variety of text analysis tools to be cascaded and applied to a set of documents.

For our purposes, we have applied natural language processing modules such as Tokeniser, Gazetteer, and Java Annotation Patterns Engine (JAPE), each module providing input to the next. The last two modules are explained further below.

In addition to these functionalities, one can also use entity extraction and syntactic parsing components. For a particular domain, it is important to provide gazetteer lists and JAPE rules. In general, there is a cascade from *lower level* information in the parts of speech and gazetteer lists to *higher level* information where lower level information is used to compose more complex units of information. As a working strategy, the lists capture simple, unsystematic patterns, leaving the JAPE rules to capture systematic, complex patterns.

Figure 1 represents the work flow (derived from the work flow diagram in (Wyner and Peters, 2010)), where an initial specification guides the definition of gazetteer lists and JAPE rules. The process cascade is applied to the corpus, which results in an annotated text. Examining the results, one determines what to modify in the gazetteer lists and JAPE rules until one achieves desired annotations. Thus, we have an *iterative process* which supports experimental refinement of the lists and rules that induce annotation.

3.1. GAZETTEER LISTS

A gazetteer is a list of lists. Each list is comprised of strings that are associated with a central concept or with some elements of the text. The lists annotate the words and strings with the MajorType of the list; they provide the bottom level of annotation on which higher level annotations are constructed using JAPE rules. The gazetteer lists discussed here are manually composed.

We initially worked with gazetteer lists from (Bransford-Koons, 2005). However, while the lists may "work", they are clearly in need of reconstruction and extension, which we discuss. One observation is that the lists are defined for US case law and particularly the California district courts. Thus, we cannot simply apply the lists to different jurisdictions, e.g. the United Kingdom; the lists and rules must be localised to different contexts. For instance, the term `Fifth Appellate District` or `Municipal Court of....` may not occur in the UK. Similar issues arise with case citations, roles of participants, causes of action, and so on. More technically, lists have alternative graphical (capital or lower case) or morphological forms, which would be bet-

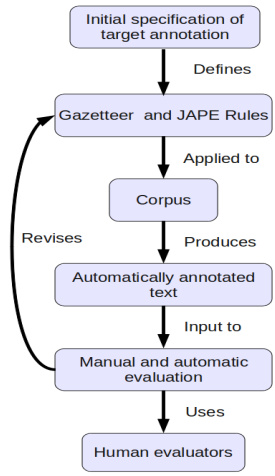

Figure 1. A Workflow Diagram

ter addressed using GATE's Flexible Gazetteer, which homogenises graphical forms and lemmatises words (providing a "root" form). As a general strategy, it is best to create lists with "unique" word forms or fixed phrases rather than those which may otherwise be constructed by JAPE rules. Taking these considerations into account, we created lists for particularly legal terminology and used the Flexible Gazetteer. The lists thus comprise a conceptual cover term; for example, a search for judgments or legal parties in a corpus will return cases and passages which contain terms found in these lists:

- judgements.lst. Terms related to judgment: grant, deny, reverse, overturn, remand,....
- legal_parties.lst. Terms for legal roles: amicus curie, appellant, appellee, counsel, defendant, plaintiff, victim, witness,....

A range of lists such as the two sampled below bear on "indicators" of structure. For example, "v." is used in cases to indicate the opposing parties, so it can be used to leverage identification and annotation of parties which appear on either side of the indicator. These are not unproblematic: the indicator might incorrectly label an abbreviated first name. There may be better ways to find judges than the initial "J."; in particular, as the list of judges is finite and give by the court system, it might be simplest to use such a list rather than applying text mining to finding it.

- legal_casenames.lst. Terms that can be used to indicate case names: v., In Re,....
- judgeindicator.lst. The indicator J.. This is a problematic indicator if it is part of an individual's name.

In other lists, we have phrases, abbreviations, and case citations. For phrases, there are two strategies. (Bransford-Koons, 2005) follows the strategy of listing the possible phrases. The alternative which we adopt is to provide bottom level lists for constituent parts of the phrases, then constructing the complex phrases by rule. The former requires a finite list; it will not annotate a novel phrase. Constructing phrases requires that the output be checked against actual phrases so it does not over generate. The treatment of abbreviations in GATE is not entirely clear, though (Bransford-Koons, 2005) simply lists them. For example, one would want to link the abbreviation with the full form, e.g. `Fifth Appellate District` and `Fifth App. Dist.`, and moreover, there may be a range of alternative abbreviations. One strategy is to have related lists - a list of phrases where the abbreviation of the phrase is a `MinorType`, and a list of abbreviations where the correlated phrase is a `MinorType`. In our view, more general solutions are better than specific ones which list information; lists ought to be contain arbitrary information, while JAPE rules construct systematic information. Case citations combine the issues of phrases, abbreviations, and alternative forms. We may have a citation such as `Cal.App. 3d` which abbreviates the California Court of Appeals, Third District. Clearly, each part is a component that can be reused in other citations. Moreover, as spaces matter in text analysis, we must account for alternatives, `Cal.App.3d` and `Cal. App. 3d`.

- lower_courts.lst. Phrases for other courts: Municipal Court of, Superior Court of,....
- legal_code_citations.lst. Code citations: Civ. Code, Penal Code,....

Some of the terms are functional; that is, both legal parties and counsel names are roles that individuals have with respect to a particular context. In one context, an individual may be a plaintiff, while in another the defendant. In annotating an individual with a functional role, e.g. an individual as plaintiff, we rely on local context within the text and do not presume that the individual's annotation applies across cases.

Finally, (Bransford-Koons, 2005) provides a range of terms which relate to the content of the case. For example, a case of criminal assault is marked by the appearance of terms bearing on weapon or intention.

- weapons.lst. A list of items that are weapons: assault rifle, axe, club, fist, gun,....

- intention.lst. Terms for intention: intend, expect,....

While it would be meaningful to index cases according to such content, they present several problems. Clearly, whether something is a weapon or criminal assault is context dependent since in some other context they might not be. How could one bound the range of relevant terms appropriately and give them interpretations that are relevant to the context? For example, isn't any object a possible weapon? These may be terms which, as discussed in (Wyner and Hoekstra, 2010), are developed in independent modules; we do not want to develop a full theory of space, time, instruments, intention, or causation.

3.2. JAPE RULES

Given the bottom-level annotations provided by the lists, we have JAPE rules which make the annotations graphically represented and available for higher level annotations. Below is a partial list of annotations given by JAPE rules.

- AppellantCounsel: annotates the appellant counsel.
- DSACaseName: annotates the case name.
- CauseOfAction: annotates for causes of action.
- DecisionStatement: annotates a sentence as the decision statement.
- JudgeName: annotates the names of judges.

Some of the JAPE rules simply translate the Lookup type into an annotation such as `Weapon`, while other rules use the Lookup type and context to annotate a text span such as `AppellantCounsel` and `DecisionStatement`. In the following sample rule, a sentence which contains a judgment term (e.g. affirm, overturn, etc) followed by a judge's name is labeled a decision statement. The rule relies on a standard format, where the case decision is followed by the judge's name; were similar patterns to appear in the case, then they too might be mis-annotated as a decision of the case.

```
Rule: DecisionStatement
Priority: 10
(
{Sentence contains JudgementTerm}
):termtemp
{JudgeName}
–>
:termtemp.DecisionStatement = {rule = "DecisionStatement"}
```

3.3. RESULTS

In this section, we give some of the results of running our GATE application over our corpus, giving the results using the graphical output of GATE

We have the following sample outputs from our lists and rules applied to *People v. Coleman, 117 Cal App. 2d 565*. The coloured highlights on the case text are associated with the same coloured annotation. We can output an XML representation to indicate the annotation. In Figure 2, we find the address, court district, citation, case name, counsels for each side, and the roles. The results give a flavour of the annotations, though further work is required to refine them.

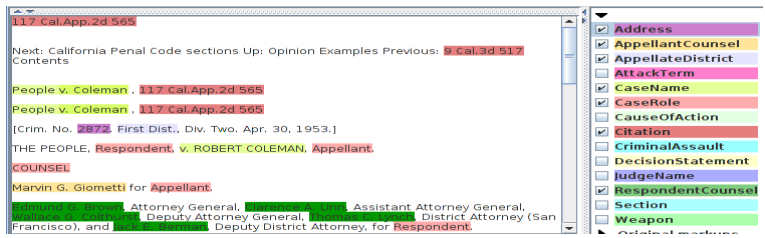

Figure 2. Case Information I

In Figure 3, we focus on additional information such as structural sections (e.g. Opinion), the name of the judge, and terms having a bearing on criminal assault and weapons. In Figure 4, we identify the decision.

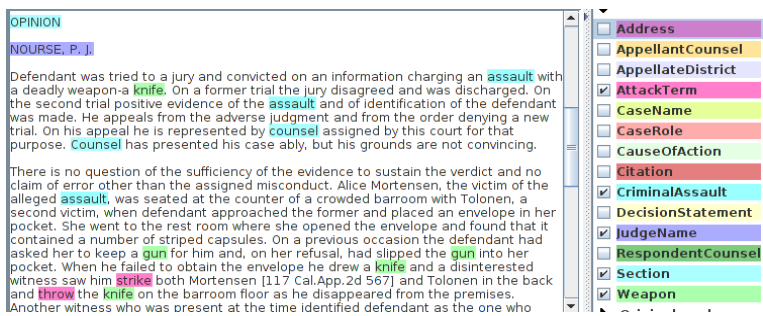

Figure 3. Case Information II

Figure 4. Case Information III

4. Conclusion

In this paper, we have outlined and extended a proof of concept approach to text mining legal cases in order to extract a range of particular elements of information from the cases. While a relatively small system applied to a very small corpus, the lists and rules approach can be extended further and relatively easily. Further developments using this approach to text mining would be to relate the extracted information to an ontology which is directly incorporated into the GATE pipeline. A second development would be to engage a wide range of users (e.g. law school students) in a collaborative, on line annotation task using GATE TeamWare. Not only would this have didactic purposes (to focus the attention of students on close analysis of the text), but it would also help to build up a body of annotated texts for further research as well as development of a gold standard that could be used for machine learning.

References

Aleven, A. (1997), *Teaching case-based argumentation through a model and examples*. PhD thesis, University of Pittsburgh, 1997.

Ashley, K. (1990), *Modelling Legal Argument: Reasoning with Cases and Hypotheticals*. Bradford Books/MIT Press, Cambridge, MA, 1990.

Ashley, K. and Brüninghaus, S. (2009), Automatically classifying case texts and predicting outcomes. *Artif. Intell. Law*, 17(2):125–165, 2009.

Bransford-Koons, G. (2005), Dynamic semantic annotation of California case law. Master's thesis, San Diego State University, 2005.

Cunningham, H., Maynard, D., Bontcheva, K, and Tablan, V. (2002), GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.

Gangemi, A., Sagri, M., and Tiscornia, D. (2005), A constructive framework for legal ontologies. In V.R. Benjamins, P. Casanovas, J. Breuker, and A. Gangemi, editors, *Law and the Semantic Web*, pages 97–124. Springer Verlag, 2005.

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

Hafner, C. (1987), Conceptual organization of case law knowledge bases. In *ICAIL '87: Proceedings of the 1st International Conference on Artificial Intelligence and Law*, pages 35–42, New York, NY, USA, 1987. ACM.

Hoekstra, R., Breuker, J., Bello, M., and Boer A. (2009), LKIF core: Principled ontology development for the legal domain. In Joost Breuker, Pompeu Casanovas, Michel C. A. Klein, and Enrico Francesconi, editors, *Law, Ontologies and the Semantic Web*, volume 188 of *Frontiers in Artificial Intelligence and Applications*, pages 21–52. IOS Press, 2009.

Jackson, P., Al-Kofahi, K., Tyrell, A., and Vachher, A. (2003), Information extraction from case law and retrieval of prior cases. *Artificial Intelligence*, 150(1-2):239–290, November 2003.

Lame, G. (2004), Using NLP techniques to identify legal ontology components: Concepts and relations. *Artificial Intelligence and Law*, 12(4):379–396, 2004.

Moens, M.-F., Uyttendaele, C., and Dumortier, J. (1997), Abstracting of legal cases: the salomon experience. In *ICAIL '97: Proceedings of the 6th International Conference on Artificial Intelligence and Law*, pages 114–122, New York, NY, USA, 1997. ACM.

Peters, W. (2009), Text-based legal ontology enrichment. In *Proceedings of the workshop on Legal Ontologies and AI Techniques*, Barcelona, Spain, 2009.

Peters, W., Sagri, M.-T., and Tiscornia, D. (2007), The structuring of legal knowledge in LOIS. *Artificial Intelligence and Law*, 15(2):117–135, 2007.

Rissland, E., Skalak, D., and Friedman, T. (1996), BankXX: Supporting legal arguments through heuristic retrieval. *Artificial Intelligence and Law*, 4(1):1–71, 1996.

Schweighofer, E. and Liebwald, D. (2007), Advanced lexical ontologies and hybrid knowledge based systems: First steps to a dynamic legal electronic commentary. *Artificial Intelligent and Law*, 15(2):103–115, 2007.

Weber, R., Ashley, K., and Brüninghaus, S. (2005), Textual case-based reasoning. *Knowledge Engineering Review*, 20(3):255–260, 2005.

Wyner, A. and Bench-Capon, T. (2007), Argument schemes for legal case-based reasoning. In Arno R. Lodder and Laurens Mommers, editors, *Legal Knowledge and Information Systems. JURIX 2007*, pages 139–149, Amsterdam, 2007. IOS Press.

Wyner, A. and Hoekstra, R. (2010), A legal case OWL ontology with an instantiation of *Popov v. Hayashi*. *Knowledge Engineering Review*, xx:xx, 2010. To appear.

Wyner, A. and Peters, W. (2010), Towards annotating and extracting textual legal case factors. In *Proceedings of the Language Resources and Evaluation Conference Workshop on Semantic Processing of Legal Texts*, Malta, 2010. To appear.