Lexical Semantics and Expert Legal Knowledge towards the Identification of Legal Case Factors

Adam WYNER [a], Wim PETERS [b]
[a] *University of Liverpool*
[b] *University of Sheffield*

Abstract. Legal case factors are textually represented facts which are represented in reported legal case decisions. Precedent decisions contribute to the decision of a case under consideration. As textually represented facts, factors linguistically encode semantic properties and relationships among the entities which can be leveraged to identify and extract the legal case factors from decisions. We integrate legal and linguistic resources in a text analysis tool with which we annotate textual passages. Using annotations tailored to legal case factors, the legal researcher can rapidly zero in on textual spans which represent specific combinations of factors, participants, and semantic properties which bear on who played what role with respect to a factor. The research reports progress on the development of a tool.

Keywords. case based reasoning, text analysis, legal factors, lexical semantics, expert knowledge

Introduction

Case based reasoning is central in *common law* legal systems, where lawyers argue a current undecided case on the basis of decided cases, which are legal precedents.[1] The lawyers compare and contrast a current undecided case against precedents with respect to the facts of the cases and the applicable laws. Based on precedents, the facts, law, and legal arguments, judges and juries decide a case. However, reasoning about the facts is complex since one must consider whether the facts are the same in each case, whether one fact subsumes another, whether one fact outweighs another, and so on. The reasoning proceeds by a counterbalancing of facts with respect to the law. Prototypical fact patterns are referred to as *factors* and the analysis of what factors hold in a precedent case is *factor analysis*. Clearly, to carry out case based reasoning, the essential first step is to determine what factors hold in reported decisions, which is different from establishing the facts of the case in the first place. As cases are reported in text, factor analysis is a form

 Corresponding author: Adam Wyner, e-mail: adam@wyner.info.

of linguistic analysis, albeit usually following an intuitive approach to language and without tool support.

Manual text annotation in general and factor annotation in particular of unstructured linguistic information is a complex, time-consuming, error-prone, and knowledge intensive task. Techniques which facilitate factor analysis would help lawyers find relevant cases. Moreover, such techniques and results contribute to research by highlighting the underlying linguistic knowledge used in legal reasoning. Using Semantic Web technologies in annotating documents, such as XML and ontologies, novel methods could be developed to analyse the law, make it more available to the general public, and to support automated reasoning. Nonetheless, the development of such technologies depends on making legal cases structured and informative for machine processing.

In this paper, we report on new semi-automated legal text analysis tool which incorporates lexical semantics and expert legal knowledge for the identification of legal case factors. First, we set the objectives. In contrast to fully automatic annotation systems [1] or machine learning approaches [2], our tool automatically enriches the text with annotations which a legal professional or research can use to *flexibly* and *interactively* search for, highlight, and extract the relevant information from the corpus. In addition, the tool developer can refine fundamental elements of the tool in order to add functionality and improve performance. Second, to a previous tool [1], which represented aspects of legal domain knowledge, we add lexical semantic linguistic information, looking especially at *thematic roles*, *selection restrictions*, and *semantic predicates*. Thus we integrates lexical semantic and expert legal knowledge in a powerful investigative tool.

The structure of the paper is as follows. In section 1, we outline background, textual source material, legal case factors, and lexical semantics. In section 2, we discuss the text analysis system we use, General Architecture for Text Engineering(GATE), and how we represent legal case factors and lexical semantic information in GATE. In section 3, sample results of applying our tool to a textual corpus are shown. Finally, in section 4, we compare our approach to fully automated and machine learning approaches. Overall, we demonstrate the feasibility and broad applicability of our tool to legal information extraction.[2]

1. Background and materials

In this section, we set the context of the research in AI and Law, discuss the corpus, outline the factors we focus on, and indicate elements of lexical semantics.

1.1. Background

For our purposes, we can identify two main branches of research on legal case based reasoning with factors in AI and Law: knowledge representations of cases and reasoning systems over a knowledge base; and textual analysis to annotate and extract the information from the linguistic representation. Clearly, the two

[2]All the materials, lists, and JAPE rules are available for testing and development under an Attribution-Non-Commercial-Share Alike 2.0 license. Contact the first author for the files.

are related. Usually, the knowledge base is derived from manual analysis of the text and represented abstractly (cf. [3], [4], [5], [6]). In [3], for example, cases are manually annotated with factors; however, the factors are informally described along with information about when they do or do not appear.

Manual analysis of cases for factors faces the knowledge acquisition bottleneck, which is the difficulty of annotating the information in textual sources. To address this, natural language processing techniques (NLP) are applied to address a range of issues such as ontologies, summarisation, precedent link extraction ([7,8,9,10], though they are tangential to our topic. More relevant is [11], which claims to apply syntactic analysis to extract events; while syntactic analysis may well support event identification in general, we are interested in factor analysis, which draws on particular *semantic* features of the text.

[12,2,1] pursue the connection between knowledge representation and text extraction. [12] develop a search in a case base with respect to a database of textual excerpts from cases that are considered to be relevant; however, the excerpts are not structured, and there is no semantic annotation. [2] apply machine learning NLP techniques such as propositional patterns to a *squib* of a case, not the source text, and classify rather than annotate text. The patterns are linguistically restricted. Moreover, it is difficult to verify the results since the materials used for the study are not publicly available. [1] propose a knowledge intensive tool for automated textual analysis of legal case factors in which elements are the *building blocks* of the base level factors. However, the results of that study positively identified fewer high level factors than expected, though the approach would lead one to expect there would be more false positive identifications. In addition, the results drew our attention to the question of just what linguistic and legal knowledge is required to reliably identify the factors.

1.2. Corpus and Factors

We derive a corpus from CATO corpus of 140 cases which are analysed with the CATO factors in [3]. These cases and factors concern intellectual property; they have been well-studied and continue to be relevant to legal case based reasoning. As not all cases from the CATO corpus are openly or freely available, we made a selection of 39 number of cases which bear on a limited range of factors.

[3] discusses a factor hierarchy comprised of base, intermediate, and higher level factors. Factors at every level are associated with the side of the case, the plaintiff or the defendant, that they support. To focus the discussion, we only consider some of the base level factors.

In [3], 27 base level factors are presented. As in the example F1, factors are represented with an index (e.g. F1), a label *Plaintiff-disclosed-information-in-negotiations*, the side the factor supports (defendant or plaintiff), a description of the factor, and comments on when the factor does and does not apply.[3]

- F1 Plaintiff-disclosed-information-in-negotiations
- Favours defendant.

[3]Here we use the edited version of the factors [3] as discussed in [1].

- Plaintiff disclosed information during negotiations with defendant. The defendant fairly obtained the information and the plaintiff was not interested to maintain the information as a secret.
- Applies if the plaintiff disclosed the information to defendant during negotiations for a joint venture, licensing agreement, sale of a business, etc..
- Does not apply if the defendant learned the information while employed by plaintiff.

The objective of the semi-automated tool in its current stage of development is to help to identify material in the text of the case decision which is associated with the factor. We do this from two directions: we decompose the base level factors into their component linguistic indicators; and, we introduce rich *lexical semantic* annotations into the text.

1.3. Term extraction and expansion

For the development of domain knowledge, we decompose each base level factor into its semantically salient terms and identify plausibly semantically related terms (synonyms); the related terms are then identified under a conceptual cover. We refer to these terms as *factoroids*, since they are not factors themselves, but are the elements which comprise the base level factors. We describe these steps.

From the factor presentations (e.g. of F1), we manually extract and lemmatise the most semantically salient terms. For example, for F1, we have:

plaintiff, disclose, product, information, negotiation, defendant, obtain, fair means

For each term (or phrase), we consulted WordNet to manually identify terms that are synonymous or related to the legally applicable sense of term. For the term *disclose*, we find the following, which are verbs and nouns that semantically indicate the concept conveyed by *disclose*:

announce, betray, break, bring out, communicate, confide, give away, impart, inform

Such lists of terms are used by GATE as described in section 2.

1.4. Outline of Verbal Lexical Semantics

We provide a motivation for and brief overview of lexical semantics (see, [13,14,15,16] among much other research in Linguistics). In linguistic research, some of the topics of interest are: similarities and differences of meaning across different sentential forms or with respect to varying lexical items; constraints on combinations of nouns and verbs; verb classifications; and inference. A significant amount of research in Linguistics is devoted to the elicitation and formalisation of this tacit knowledge from speakers. Such knowledge is problematic for search engines unless rich annotations are appended to the linguistic form.

In Linguistics, we find the following sorts of annotations (among others):

- Syntactic - subject, verb, object, indirect object, noun phrase, verb phrase, etc. In *The plaintiff received the information from the defendant.*, *The plaintiff* is the subject.

- Thematic roles - Agent, Theme, Recipient, and others, which are the semantic properties that arguments in the sentence bear. In *The plaintiff received the information from the defendant*, *The plaintiff* is the Agent (doer of action) and *the defendant* is the Recipient.
- Selection restrictions - abstract, body_part, animate, organisation, etc. These restrict arguments bearing thematic roles to certain sorts. The sentence *The rainstorm received the information from the defendant* is odd since the subject is not an animate agent.
- Semantic predicates – command, confront, forbid, group, etc. These are abstract semantic properties of classes of verbs and which support inference. In *The plaintiff received the information from the defendant*, the semantic representation of the verb *receive* includes the semantic predicate *has_possession* which holds after execution of the action and with respect to the two participants.

The annotations are used to classify verbs according to similar semantic properties and syntactic behaviour (alternative syntactic forms with related meanings, known as diathesis alternations); indeed a key achievement has been the analysis of patterns and structure within the lexicon using such syntactic and semantic elements [14].

Annotating a text with respect to legal domain information and lexical semantics greatly enhances the capacity to highlight and extract relevant textual passages which vary in form, but convey similar or related meanings. Since legal case factors are prototypical descriptions of facts that may appear in different forms, using lexical semantic information would help to identify factors.

For our purposes, we base our work on VerbNet, which is a large, open source, on-line verb lexicon for English that represents syntactic and semantic information [15,16].[4] VerbNet gives a lexical description which applies to an exhaustive listing of verbs which adhere to the description. Annotating text with VerbNet information enables us to query a corpus for fine-grained, lexical semantic information that relates verbs of similar meanings and arguments in related roles. In the next section, we describe the implementation of our analysis of factor descriptions lexical semantics in the GATE architecture.

2. Method in GATE

In this section, we discuss how the factoroids and lexical semantics in GATE.

2.1. GATE

GATE is a framework for language engineering applications, which supports efficient and robust text processing [17]. GATE is an open source desktop application written in JAVA that provides a user interface for professional linguists and text engineers to bring together a wide variety of natural language processing tools and apply them to a set of documents. The tools are formed into a pipeline (a

[4]We acknowledge the VerbNet 3.1 license in our GATE files.

sequence of processes) such as sentence splitters, tokenisers, part-of-speak tagger, morphological analyser, gazetteer lists, and Java Annotation Patterns Engine (JAPE) rules. Less familiar, a gazetteer is a list of words that are associated with a central concept, e.g. *disclose* illustrated in section 1.3. JAPE rules are transductions which take annotations and regular expressions as input, and produce annotations as output.

Once a GATE pipeline has been applied to a corpus, the annotations can be queried through the ANNIC (ANNotations In Context) corpus indexing and querying tool [18], which allows the evaluator to enter search patterns over text annotations and to detect occurrences of semantic entities, patterns and relations at a fine-grained text level.

2.2. Gazetteers and JAPE Rules

In 1.3, we found a list of words that were related to "disclosure". These words are made into a text file, a gazetteer list, such as `disclosure.lst`, which is used by the gazetteer; the gazetteer associates the gazetteer list with majorType `disclosure`. This means that during the *lookup* phase of processing, when the gazetteer lists are consulted, terms (i.e. Tokens) which appear on a list are annotated with Lookup as the majorType from the relevant list. For example, suppose a text with a token such as "confide" within it; when GATE finds this token during the lookup phase, GATE annotates the token with Lookup = `disclose`. If we subsequently *query* for tokens with Lookup = `disclose`, the results include the token "confide" among other tokens with the same Lookup. Thus, the function of the gazetteer lists is to provide a *cover concept* for related terms that can be queried or used by subsequent annotation processes.

In addition, we can use the Lookup value to create JAPE rules, which creates annotations that are visible as highlighted text and searchable in ANNIC. For instance, given the Lookup majorType `disclosure`, we create a rule to annotate text with `DisclosureFactoroid`, which can be highlighted with a colour; every factoroid from the gazetteers and JAPE rules is suffixed with *Factoroid*.

In the implementation, we have 40 gazetteer lists for legal domain knowledge along with their related JAPE rules among them:

> UseHave, Confidential, Disclosure, Disseminate, FormEmployee, Hire, Information, Know, Negotiate, Outsider, SecureInformation, LegalParties

While the legal domain knowledge and general knowledge may overlap (e.g. *hire* as a legally relevant term and also as a general knowledge verb with lexical semantic information), we allow this redundancy since each contributes different and useful information.

VerbNet is available as a downloadable folder of XML files which represent verb classes, the elements of the classes, the lexical description, the semantic frame, and any alternative lexical descriptions or semantic frames.[5] Instead of using the VerbNet API, we have converted VerbNet information into GATE gazetteers, which means that verbs in the corpus are annotated with respect to VerbNet information. Every verb in GATE's version of VerbNet has associated

[5] `http://verbs.colorado.edu/~mpalmer/projects/verbnet.html`

with it the general class to which in belongs as well as particular lexical semantic descriptions in terms of thematic roles, selection restrictions, and semantic predicates. We have JAPE rules to create annotations for all the thematic roles, selection restrictions, and semantic predicates used by VerbNet. With the annotations, we can query for features or write JAPE rules. The searches and JAPE rules can be as fine-grained or complex as the VerbNet information allows. For clarity, the 30 thematic roles are suffixed with *ThematicRole*, the 36 selection restrictions on arguments with *SelectRestrict*, and the 144 semantic predicates with *SemanticPred*. The annotations are the building blocks of a language for complex searches or compound JAPE rules.

3. Results of Queries

Once the corpus is annotated, we can query the database with respect to the annotations created by the linguistic and conceptual analysis above. The query language of GATE is highly expressive, allowing us to search for different patterns of legal domain and lexical semantic information. We describe an example search path to exemplify the approach.

As part of our investigation into whether F1 (F1 Plaintiff-disclosed-information-in-negotiations) applies, we can seek an answer to the questions related to this factor such as who disclosed or failed to disclose what? For this purpose, we formulate the following general query in ANNIC in order to retrieve relevant instances from the cases:

Step 1: Single term query – {*LegalPartiesFactoroid*}

The results are textual passages (reported in ANNIC) which contain annotated legal roles that particular individuals adopt. We can search further with respect to the context in which the annotations appear.

The inspection of the context in ANNIC will give suggestions for possible further refinements of the general query. For instance, considering the factor information (e.g. for F1 described above) and previous results, we find that neglecting to disclose appears commonly; we explore how, where, and by whom any neglect has taken place, which may bear on F1 or some other factor. The richness and semantic nature of our annotations allows us to adopt a step-wise query refinement:

Step 2: Simple collocations query –

{*LegalPartiesFactoroid*} *(*{*Token*}*)*3* {*NeglectSemanticPred*}

This means that if a legal party and a "neglect" semantic predicate co-occur in the text with up to 3 tokens in between, we may have a valid instance of what we are querying for; if we want to query for the plaintiff alone, we can exchange *LegalPartiesFactoroid* with *PlaintiffFactoroid*.

We can now further refine the query by adding the *DisclosureFactoroid* to the query, which is constituted by either the "disclose" semantic predicate added by VerbNet, or the "disclosure" class obtained from WordNet. The search can be further refined with the *InformationFactoroid*:

Step 3: Complex collocations query –

{*LegalPartiesFactoroid*} *(*{*Token*}*)*3* {*NeglectSemanticPred*} *(*{*Token*}*)*5*

{*DisclosureFactoroid*} ({*Token*})*3{*InformationFactoroid*}
Figure 1 shows the results for this complex collocations query. It is displayed as a context snippet including the highlighted text span that matches the query, and its factoroid components represented as blocks covering text elements. The query returned three additional results.

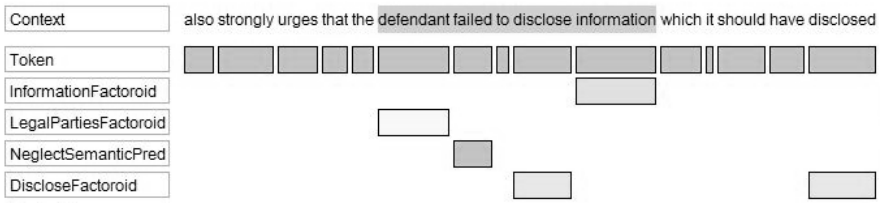

Figure 1: Graphical display of query results

Given tokens which intersperse between the components, textually dispersed collocations can be located. Thus, while factors are prototypical fact patterns, their textual realisation may be rather different, where the factoroids can be dispersed over the paragraph in different sentences. This dispersal of information may be one cause of the poor results for base level factor identification reported in [1]. Generally, the greater the proximity and density of the annotations, the more likely it is that the passage bears on the base level factor F1.

Further information for inspection can be gleaned from other annotation types. For instance, we can consult VerbNet for additional information on semantic roles associated with semantic predicates such as Agent, Patient, and Beneficiary; alternatively, we can identify the VerbNet semantic predicates from other verbs in the textual construction.

4. Discussion and Conclusion

Given the intricacies and unique characteristics of legal knowledge, we opt for the creation of a tool for the manual identification and evaluation of legal factors, collaboratively assisted by the automatic annotation of legal texts on the basis of NLP extensible resources and algorithms implemented in GATE. This will allow interested parties to establish through empirical exploration and verification:

- Whether the NLP information is correct and useful for knowledge capturing and algorithm formation;
- What the relations are between elements at the same level of the analysis (e.g. the syntactic and semantic relations between words) and between the levels of the analysis (e.g. whether the factoroids are constitute or merely indicate a base level factor).

This approach, which involves a collaborative, incremental approach wherein human experts are assisted by resources and rules, is an interesting avenue to investigate. The collaborative aspect extends to the human experts themselves,

since the manual identification process can be performed by multiple experts, and therefore function as a bottom-up consensus building methodology for case interpretation. In addition, the system can contribute to the development of a gold standard or fully-automated system. The output of the approach can be evaluated against the results of on-line collaborative annotation systems such as GATE TeamWare [1]. However, as an interactive, semi-automatic system, we do not develop a gold standard corpus or apply machine learning. We discuss and justify our position with respect to legal texts.

Gold standards are necessary for machine learning, forming the initial corpus to which the learning algorithm is applied. Yet the creation of gold standards is an intrinsically difficult task, representing the consensus interpretation of the corpus by the evaluators some protocol for annotating the text [19], in effect, encoding the expert knowledge of the domain. Inter-annotator disagreement is common and problematic since the interpretation may vary with respect to the texts and domain. Moreover, there is a presumption that the gold standard represents a homogeneous interpretation which is also to be found in the test texts. This may not be so, especially given a corpus of texts which themselves are highly interpretive. Corpora of legal cases may be less homogeneous than corpora in other domains where text analytics are successfully applied such as on chemistry. In our approach, we make explicit all the *knowledge* which is used in annotating the text, thus representing the information that gold standard evaluators use in making the standard.

With respect to machine learning, there are functional and scientific issues. With respect to the functional issue, to maintain judicial authority, current legal case based reasoning requires the citation to be explicit, verifiable, and justified; that is, not only must the cited passage be identified, but its relevance and meaning must be analytically supported and grounded in linguistic expressions. Our approach to lists and rules supports just such explicit, verifiable, and justified citations. This contrasts with other areas of legal reasoning, where proximate reasoning is allowed: in reasoning about evidence, *proof standards* allow degrees of proof; and, in reasoning about the applicability of a law, vague and arguable concepts may be used. Given this, the question is whether the results of machine learning in the identification of factors and in relating cases would suit the explicit, verifiable, and justified requirements of legal case based reasoning. We think not since the justification of the classification of a factor or the relationship between cases is opaque and approximate.

In relation to the scientific issue, we are interested in what is *knowledge of the law* that legal professionals have. Crucial issues of legal knowledge are identification of the elements and their combinations which contribute to legal reasoning along with justification, explanation, transparency, and textual traceability. Current approaches to machine learning do not satisfy these issues. Our bottom up, knowledge intensive approach does, though it is initially more limited.

There are many areas for future investigation such as identifying a range of useful search patterns in the law and elaborating on the legal knowledge that is represented in the tool. In order to further automate factor annotation, we must address the problems of word sense selection and the mapping of lexical semantics to syntactic arguments. Also, we intend to integrate other linguistic

resources such as syntactic parsing, ontologies, and other existing linguistic and terminological knowledge bases. We will look into some level of application of machine learning approaches. Finally, the interface and results ought to be made more user friendly.

References

[1] Wyner, A., Peters, W.: 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.

[2] Ashley, K.D., Brüninghaus, S.: Automatically classifying case texts and predicting outcomes. Artificial Intellence and Law **17**(2) (2009) 125–165

[3] Aleven, V.: Teaching case-based argumentation through a model and examples. PhD thesis, University of Pittsburgh (1997)

[4] Rissland, E.L., Ashley, K.D., Branting, L.K.: Case-based reasoning and law. The Knowledge Engineering Review **20** (2006) 293–298

[5] Wyner, A., Bench-Capon, T.: Argument schemes for legal case-based reasoning. In Lodder, A.R., Mommers, L., eds.: Legal Knowledge and Information Systems. JURIX 2007, Amsterdam, IOS Press (2007) 139–149

[6] Wyner, A.: An ontology in OWL for legal case-based reasoning. Artificial Intelligence and Law **16**(4) (2008) 361–387

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

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

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

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

[11] Maxwell, K.T., Oberlander, J., Lavrenko, V.: Evaluation of semantic events for legal case retrieval. In: ESAIR '09: Proceedings of the WSDM '09 Workshop on Exploiting Semantic Annotations in Information Retrieval, New York, NY, USA, ACM (2009) 39–41

[12] Daniels, J.J., Rissland, E.L.: Finding legally relevant passages in case opinions. In: ICAIL '97: Proceedings of the 6th International Conference on Artificial intelligence and Law, New York, NY, USA, ACM (1997) 39–46

[13] Hayes, B., Curtiss, S., Szabolcsi, A., Stowell, T., Stabler, E., Sportiche, D., Koopman, H., Keating, P., Munro, P., Hyams, N., Steriade, D.: Linguistics: An Introduction to Linguistic Theory. Wiley-Blackwell (1999)

[14] Levin, B.: English Verb Classes and Alternations: A Preliminary Investigation. University of Chicago Press (1993)

[15] Kipper, K., Korhonen, A., Ryant, N., Palmer, M.: A large-scale classification of english verbs. Language Resources and Evaluation **42**(1) (2008) 21–40

[16] Palmer, M., Hwang, M., Hwang, J., Brown, S., Schuler, K., Lanfranchi, A.: Leveraging lexical resources for the detection of event relations. In: Proceedings of the AAAI Spring Symposium on Learning by Reading, Stanford, CA (2009)

[17] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: 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)

[18] Aswani, N., Tablan, V., Bontcheva, K., Cunningham, H.: Indexing and querying linguistic metadata and document content. In: Proceedings of 5th International Conference on Recent Advances in Natural Language Processing, Borovets, Bulgaria (2005)

[19] Baron, J., Lewis, D., Oard, D.: TREC-2006 legal track overview. In: The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings. (2006)