

Towards Annotating and Extracting Textual Legal Case Factors

Adam Wyner¹, Wim Peters²

¹University College London, ²University of Sheffield
adam@wyner.info, w.peters@dcs.shef.ac.uk

Abstract

Case based reasoning is a crucial aspect of *common law* practice, where lawyers select precedent cases which they use to argue for or against a decision in a current case. To select the precedents, the relevant facts (the case factors) of precedent cases must be identified; the factors predispose the case decision for one side or the other. As the factors of cases are linguistically expressed, it is useful to provide a means to automate the identification of candidate passages. We outline and report the results of our approach to the identification of legal case factors which follows a bottom-up knowledge heavy strategy and uses the General Architecture for Text Engineering system. Salient lexical items are selected, concept classes of related terms are created, and annotation rules for simple and compound concepts are provided. The annotated concepts can be extracted from the cases, and cases can be classified with respect to the concepts. In addition to supporting extraction of relevant information, the approach has a didactic use in helping to train lawyers to perform close textual analysis. Finally, we carry out an initial collaborative, online annotation exercise using GATE TeamWare in order to develop a gold standard.

1. Introduction

Case based reasoning is a crucial aspect of *common law*, where lawyers argue a current undecided case on the basis of legal precedents, which are decided cases drawn from a legal case base.¹ The lawyers compare and contrast the current undecided case against decided cases in terms of the facts of the cases and the applicable laws. Based on the facts and arguments, judges and juries decide a case, guided by a conservative principle of *stare decisis*, which obliges the decision to be consistent with decisions of previous cases *ceteris paribus* (though sometimes decisions are overturned); where the facts of the cases vary, the lawyers and judges reason with respect to a counterbalancing of factors and their role in the law and society. Prototypical fact patterns are referred to as *factors* and the analysis of the factors in a case is *factor analysis*; a given factor may predispose the case to be decided in favour of one side or the other of the dispute. For instance, in the domain of *intellectual property* cases, where a plaintiff claims a defendant stole the plaintiff's intellectual property, a factor would be whether or not the plaintiff required the defendant to sign a non-disclosure agreement prior to disclosing the secret. If the plaintiff did not require the agreement, this fact would predispose the decision in the case in favour of the defendant since, after all, the lack of a requirement indicates that the plaintiff was negligent in identifying and protecting his property. On the other hand, if the plaintiff did require the agreement, but the defendant did not abide by it, then this fact would predispose the decision in favour of the plaintiff since the plaintiff was making efforts to protect his property, but the defendant violated the agreement. The actual outcome of the case depends on the full range of factors, their relationships, the law, and procedural moves by the lawyers, among other aspects that contribute to a court decision. Thus, it is crucial to determine what factors hold of a case as reported in the language of the case decision (which is distinct from determining the facts in the first instance), both for research and in practice.

The goal of this paper is to demonstrate the feasibility of applying automated tools to support the identification of factors and to annotate them for subsequent information extraction and processing. Text annotation in general and factor annotation in particular of unstructured linguistic information is a complex, time-consuming, error-prone, and knowledge intensive task; it is a difficult aspect of the "knowledge acquisition bottleneck" in information processing (Forsythe and Buchanan, 1993). Techniques which facilitate factor analysis would help lawyers find relevant cases. In addition, by using Semantic Web technologies 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 work, the semantic annotations are the leaves of a hierarchy of factors, where the leaves indicate higher level factors to a lesser or greater extent, rather than precisely. In general, factors constitute conceptual entities in legal discourse, which can be of various levels of semantic complexity. At the lowest level, factors can be regarded as similar to linguistic expressions such as domain-specific nominal, adjectival, or verbal terms and keywords. These combine into increasingly complex higher level factors such as collocations or verbal predicates. The workflow we are aiming at accommodates all factor levels by annotating higher level factors in terms of lower level constituents. It allows an incremental bridging of levels by means of the addition of fine-grained domain-specific patterns of language use, in whichever linguistic form. In order to provide the initial linguistic building blocks to bridge levels of linguistic description, we apply text preprocessing steps such as sentence splitting, tokenisation, part-of-speech tagging, and lemmatisation. The flexible combination of these building blocks makes possible the mapping of the surface language onto the underlying conceptual factor organization of the legal case domain. This initial study will lead to further research, where we will iteratively refine the annotations to more closely approximate the linguistic realisations.

¹2010 ©Adam Wyner and Wim Peters. Corresponding author: Adam Wyner, adam@wyner.info.

In this paper, we apply natural language information extraction techniques to a sample body of cases, which are unstructured text, in order to automatically identify and annotate the factors. Annotated factors can then be extracted for further processing. Not only does such an approach offer to save time and money, but it also reveals key elements at the basis of legal case based reasoning and advances research in AI and Law. In section 2., we first outline some background and the materials. In section 3., we detail the methodology, which uses the General Architecture for Text Engineering(GATE) system, and the results of our method. In section 4., we outline a manual annotation experiment using GATE TeamWare, which allows comparison to the automatic annotation, and the results of the experiment. In section 5., we compare our approach to the key previous approach to factor extraction. Finally, in section 6., we summarise our report and outline future work to improve our results. Overall, we demonstrate the feasibility of our approach and the opportunities for open source, collaborative refinement.²

2. Background and materials

2.1. Background

Legal case based reasoning with factors has long been a research area in AI and Law. For our purposes, we can identify two main branches of research. One branch develops knowledge representations of cases and reasoning systems over a knowledge base. While the knowledge base may be derived from a textual case base, most often this is done by manual analysis, where the knowledge representation abstracts from the text and the reasoning rules apply to the abstract elements of the knowledge base (cf. (Hafner, 1987), (Ashley, 1990), (Rissland et al., 1996), (Aleven, 1997), (Chorley, 2007), (Rissland et al., 2006), (Wyner and Bench-Capon, 2007), (Wyner, 2008)). However, this line of research does not address the knowledge bottleneck. The other branch attempts to address the bottleneck with textual analysis – the annotation and extraction of information from its linguistic realisation – using NLP techniques for ontology construction ((Lame, 2004), (Maynard et al., 2008), and (Peters, 2009)), text summarisation ((Moens et al., 1997) and (Hachey and Grover, 2006)), and extraction of precedent links (Jackson et al., 2003). However, these are tangential to our topic. Somewhat more relevant is (Maxwell et al., 2009), where events are extracted using part-of-speech tags, heads of arguments of predicates, and syntactic dependency structures; such a technique might be applicable to the identification of some factors, though that is not the object of study in (Maxwell et al., 2009).

While factor analysis and factor reasoning is of long practice in the law, formal, automated approaches are relatively more recent ((Ashley, 1990) and (Aleven, 1997)). In the CATO system of (Aleven, 1997), a case base is manually analysed, and factors are associated with the cases. CATO provides as well automated means to support reasoning about the cases with respect to the cases in order to propose

a decision. Current versions of CATO provide a system for students to index cases and argue about them (Aleven, 2003).

Figure 1 is an example from (Aleven, 1997) where students are presented with a case, *Mason v. Jack Daniel Distillery*, a list of potential factors such as *Security Measures* and *Unique Product* among others, and guidance on how to identify the factors in the text. When the factor is identified, a note is made alongside the text.

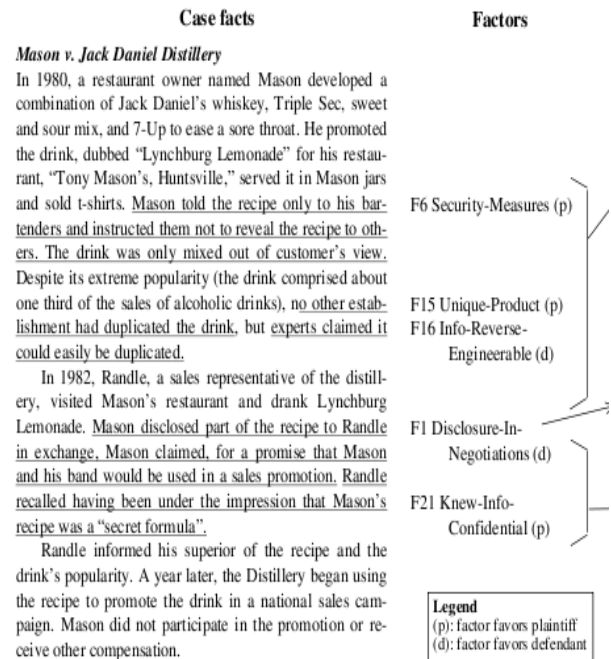


Figure 1: A case with associated factors

While the textual elements are associated with the factors, and cases are thereby indexed with respect to the factors, the association is manual and the result is not an annotation since the factor note does not mark the text directly. However, just what constitutes a factor is not formally defined, but informally given as a description along with indications of when the factor does and does not hold. One of the questions our research highlights is the structure of factors – are they schemes, events, or frames? Moreover, just what is the relationship between the lowest level linguistic indicators and higher level compound concepts?

2.2. Materials

For materials, we have drawn from the CATO corpus of cases and the CATO factors in (Aleven, 1997). Our reason is that this is a narrowly defined set of cases and factors; moreover, it is a well-studied and well-developed domain, so integrating our presentation in the context of ongoing work. As such, we can leverage the previous results and compare our results to them. Furthermore, by gathering and annotating the cases, the CATO case base can be made available to a wider range of researchers for experimentation.

The CATO corpus is comprised of some 140 cases concerning intellectual property. However, all legal case decisions

²All 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 not yet openly or freely available. Of the 140, we have gathered 39 which are available. Of these, we have selected four to work with in order to narrow the scope of the current project; we discuss the rationale for our selection below. These cases are:

- FMC Corp. v. Taiwan Tainan Giant Ind. Co, Ltd, 730 F.2d 61 (2nd Cir.1984) (FMC)
- Goldberg v. Medtronic, 686 F.2d 1219 (7th Cir. 1982) (Gold.)
- Midland-Ross Corp. v. Yokana, 293 F.2d 411 (3rd Cir.1961) (Mid.)
- Trandes Corp. v. Guy F. Atkinson Co., 996 F.2d 655 (4th Cir.1993) (Tra.)

(Aleven, 1997) discusses 27 *base level factors*, which are distinct from the intermediate and higher level factors. Factors are associated with the side of the case that they support, either the plaintiff or the defendant. For instance, if the plaintiff required the defendant to sign a non-disclosure agreement, this is a plaintiff factor since it indicates that the plaintiff was taking due measures to protect intellectual property. If the defendant learned of the intellectual property in a public forum, this is a defendant factor since it indicates that the defendant did not misappropriate the plaintiff's property. We only discuss the base level factors for these are most closely associated with the linguistic factor indicators of the text, and the intermediate and higher level factors are inferred from the base level factors. Of the 27, we have investigated the following six factors:³

Pro Plaintiff Factors

- F6 Plaintiff-adopted-security-measures
- F7 Defendant-hired-plaintiff-employee
- F21 Defendant-knew-information-confidential

Pro Defendant Factors

- F1 Plaintiff-disclosed-information-in-negotiations
- F10 Plaintiff-disclosed-information-to-outsiders
- F27 Plaintiff-disclosed-information-in-public-forum

The rationale for the selection of cases and factors is as follows. We only have a fragmentary list of the factors which appear in the cases available to us ((Aleven, 1997) and (Chorley, 2007)). We want to find at least one plaintiff factor and one defendant factor in each case, with some factors appearing in more than one case, though we have not done an analysis with respect to every factor in this set of cases. In particular, we find the following, which also indicates the winning side, where we only include those factors under investigation:

- FMC Outcome: Plaintiff

– Pro Plaintiff: F6, F7

– Pro Defendant: F10

- Goldberg Outcome: Plaintiff

– Pro Plaintiff: F21

– Pro Defendant: F1, F10, F27

- Midland Outcome: Plaintiff

– Pro Plaintiff: F7

– Pro Defendant: F10, F27

- Trandes Outcome: Plaintiff

– Pro Plaintiff: F4, F6

– Pro Defendant: F1, F10

The objective of the automated and manual annotation tasks is to automatically or manually identify material in the text of the case which is associated with the factor. We then compare and contrast the results. The manually annotated cases, suitably refined and expanded, provides a *gold standard* against which to evaluate the automated techniques. The development of both annotation approaches allows us to iteratively develop the overall objective of a well-developed factor analysis for this set of legal cases.

3. Methodology

In this section, we outline our methodology for developing the annotations, then report and discuss the results.

3.1. GATE

The techniques described in this paper rely on the GATE architecture (Cunningham et al., 2002). GATE is a framework for language engineering applications, which supports efficient and robust text processing. Overall, the GATE platform consists of two main functionalities:

- GATE Developer 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 text analysis tools and apply them to a document or set of documents. GATE Developer incorporates many NLP tools as plug-ins. Some have been developed in-house, others have been written specifically for GATE and others have been ported from stand-alone open-source tools.
- GATE TeamWare is a web-based management platform for collaborative annotation and curation. It delivers a multi-function user interface over the internet for viewing, adding and editing text annotations. It allows the specification, managing and monitoring of the workflow of the collaborative text annotation work over the internet, and structures the contributions from different actors (human and machine) into clearly-defined roles.

For our purposes, we have applied the following modules in order to our texts, each module providing input to the next; the last two modules are explained further below:

³We have maintained the numbering of the factors, but changed the labels in order to make them more informative for the manual annotation task.

- Sentence splitter, which splits the text into sentences.
- Tokeniser, which identifies basic 'tokens' or words in the text.
- Part of speech tagger, which associates tokens with parts of speech such as noun, verb, and adjective.
- Morphological analyser, which lemmatises the tokens to provide words in their *root* form. This allows us to work with uniform word forms rather than taking into consideration morphological variants as in *sing*, *sung*, and *sang*.
- Gazetteer, which is a list of lists, where each list is comprised of words that are associated with a central concept.
- Java Annotation Patterns Engine (JAPE), which enable rules to be written with annotations and regular expressions as input, and annotations as output.

In the next sections, we detail first the construction of the gazetteer lists and JAPE rules followed by results. Then we present how we worked with GATE TeamWare and our results. In Figure 2, we represent the workflow .

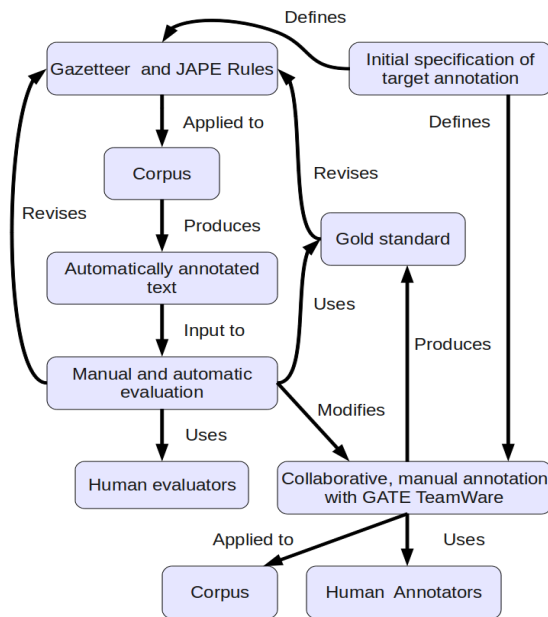


Figure 2: A Workflow Diagram

3.2. Development of GATE elements

Given our materials, the method we employed is knowledge heavy, bottom up, and cascading; we focus on the development of gazetteer lists and JAPE rules. It is knowledge heavy in the sense that in making the lists and rules, we have taken salient concepts from descriptions of case factors, used information about word relationships, and provided for alternative orders of terms. We have taken the descriptions of the case factors in (Aleven, 1997), identified

key concepts that relate to the factor, used WordNet to identify semantically related terms, then used that list of terms to define rules that provide for bottom level concept annotations. These bottom level annotations are then be used to define rules for compound annotations. We use tools in GATE to view or extract the occurrences of the annotations. As pointed out earlier, the annotations are taken to be indicative of the factors to a lesser or greater extent, where the factors are events or topics which are linguistically expressed. By iteratively refining the lists and rules, our automated processing will, we expect, approximate manual identification of the precise linguistic realisations.

3.3. A sample factor description

A sample factor presentation from (Aleven, 1997, p. 242) follows. As discussed, we are only considering the base level factors. The factor presentation contains the index (F1), a label `Disclosure-In-Negotiations`, the side favoured if the factor holds (d represents defendant and p plaintiff), a description comprised of the event or situation along with some explanatory meaning relevant to the case, and some indications of when the factor does and does not apply (which are not always given for every factor).

- F1 Disclosure-In-Negotiations (d)
- Description: Plaintiff disclosed its product information in negotiations with defendant. This factor shows that defendant apparently obtained information by fair means. Also, it shows that plaintiff showed a lack of interest in maintaining the secrecy of its information.
- The factor applies if: Plaintiff disclosed the information to defendant in the context of negotiating a joint venture, licensing agreement, sale of a business, etc.
- The factor does not apply if: Defendant acquired knowledge of plaintiffs information in the course of employment by plaintiff.

Other factor presentations are similar.

3.4. Manual term extraction

From the factor presentation, we have manually extracted the most salient terms and simplified the presentation (primarily for use in the manual annotation task). For instance, we extract the following lemmatised terms and phrases from F1:

plaintiff, disclose, product, information, negotiation, defendant, obtain, fair means, show, lack of interest, maintain, secrecy, joint venture, licensing agreement, sale of a business, acquire, knowledge, employment

We simplified the presentation of the factor:

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

3.5. Expansion of terms to create gazetteer

From the extracted terms and phrases of the factor presentations, we made classes of synonymous terms. Then we consulted WordNet to identify synonymous or salient terms that relate to some legally applicable concept. For example, for “disclosure”, we found the following:

announce, betray, break, bring out, communicate, confide, disclose, discover, divulge, expose, give away, impart, inform, leak, let on, let out, make known, pass on, reveal, tell, announcement, betrayal, communication, confidence, disclosure, divulgence, exposure

These terms comprise the strings in a gazetteer list, `disclosure.lst` with `majorType disclosure`. This means that during the lookup phase of processing, the gazetteer lists are consulted, and terms (i.e. Tokens) which appear on a list are annotated with `Lookup` as the `majorType` from the relevant list; that is, when GATE finds a token such as “confide” in the text, GATE annotates the token with `Lookup = disclose`. Thus, the function of the gazetteer lists is to provide a *cover concept* for related terms that can be used by subsequent annotation processes. As an initial development, this manual method can be used to *seed* automated methods to identify further relevant terms; alternatively, other resources can be drawn on to elaborate or refine the underlying lists. However, we do not presume prescriptive automation, where the content of the lists is fixed by the authors; rather, the lists will be refined and elaborated in a process of community development. In addition, list development may be related to ontological development, where the major type serves as a concept cover term for terms that may vary in their lexical semantics. The context dependent interpretation of lexical items is a significant problem. In legal cases, we have terms that have a functional role. For example, whether an object is a weapon or “just” an object (such as a pen) depends on the context and the actions; similarly, individuals and organisations have a functional role, an individual may be a plaintiff in one case and a defendant in another. This is often a problem in dealing with natural objects versus socially defined objects as well as when we are dealing with fixed versus flexible reference. For our purposes, we put these significant issues aside in order to begin to develop the means to identify factors; our view is that we can better address functional roles where we manually provide annotated information and reason with the information in an ontology. Finally, there are issues of polysemy of terms. However, we are addressing a highly restrictive domain (reports of decisions in case law) rather than an entirely open domain. Moreover, in a legal context, it is crucial to disambiguate terms and keep the interpretation of terms fixed. Thus, we believe these issues, while important to attend to where they occur, are not salient in our domain.

3.6. The bottom level annotation from a JAPE rule

Once Tokens have a `Lookup` value, we create JAPE rules for each `Lookup` value, which creates annotations that appear in GATE’s annotation set. For instance, given the `Lookup majorType disclosure`, we create an annotation `Disclosure`.

```
Rule: DisclosureFactor01
({Lookup.majorType == ``disclosure``})
):temp
-->
:temp.Disclosure = {rule =
``DisclosureFactor01``}
```

The annotations are the building blocks of a language for compound JAPE rules which annotate phrases or sentences with respect to two or more basic annotations.

3.7. Compound rules

In the following, we have an example compound JAPE rule annotates sequences of tokens with the `Disclosure` annotation, followed by zero or more `Tokens` within a sentence (given by `{Token, !Split}*`), followed by the `Information` annotation. The whole text span is annotated `DisclosureInformationXY`.

```
Rule: DisclosureInformationXY
({Disclosure}
({Token, !Split})*
{Information}
):temp
-->
:temp.DisclosureInformationXY = {rule =
``DisclosureInformationXY``}
```

The linear order of the annotations is crucial: in the rule above, we can only find `Disclosure` followed by `Information`; this can appear where we have, for example, an active sentence such as *Bill disclosed the information*. However, were we to have some alternative order of the annotations, then the rule above would not succeed. Therefore, we must write another rule to take into account the alternative order, where `XY` as above indicates one order, `YX` indicates another order. For every pair of annotations we want to annotate as a compound, we need at least two rules; for a rule containing 3 elements, we might require 6 rules for all the alternative orders; however, in practice, this is not clearly required. Once we have all the alternative orders, we write a “cover” rule, which makes the order irrelevant as in:

```
Rule: DisclosureInformation
({DisclosureInformationXY} |
{DisclosureInformationYX}
):temp
-->
:temp.DisclosureInformation = {rule =
``DisclosureInformation``}
```

3.8. Factor rules

Using either bottom level or compound level annotations, we define the highest level annotation rule which is intended to annotate a factor. For example, for the target factor F1 Plaintiff-disclosed-information-in-negotiations, along with annotations for Disclosure, Information, and Negotiation, we provide a rule for one order of the bottom level annotations:

```
Rule: DisclosureInformation-
NegotiationXYZ
({Disclosure}
({Token, !Split})*
{Information}
({Token, !Split})*
{Negotiation}
):temp
-->
:temp.DisclosureInformationNegotiation-
XYZ = {rule = ``DisclosureInformation-
NegotiationXYZ``}
```

We create rules for all relevant alternative orders and provide a “cover” rule such as the following for DisclosureInformationNegotiation, which for Factor F1:

```
Rule: DisclosureInformationDisseminate
( {DisclosureInformationDisseminate-
TempZYX} |
{DisclosureInformationDisseminate-
TempZXY} |
{DisclosureInformationDisseminate-
TempYXZ} |
{DisclosureInformationDisseminate-
TempYZX} |
{DisclosureInformationDisseminate-
TempXYZ} |
{DisclosureInformationDisseminate-
TempXZY}
):temp
-->
:temp.DisclosureInformationDisseminate
= {rule = ``DisclosureInformation-
Disseminate``}
```

3.9. Additional gazetteer lists and factor rules

In addition, we have 37 gazetteer lists along with their related JAPE rules. We have the list name, sample elements, and the annotation.

- usehave.lst: have, use, adopt: UseHave
- confidential.lst: confidential: Confidential
- disclosure.lst: disclosure: Disclosure
- disseminate.lst: disseminate: Disseminate
- form-employee.lst: formemployee: FormEmployee

- hire.lst: hire: Hire
- information.lst: information: Information
- know.lst: know: Know
- negotiate.lst: negotiate: Negotiate
- outsider.lst: outsider: Outsider
- secureinfo.lst: secureinfo: SecureInformation

We have factor rules constructed from this language and homogenising over word order:

- DisclosureInformationNegotiation:
F1 Plaintiff-disclosed-information-in-negotiations
- DisclosureInformationOutsider:
F10 Plaintiff-disclosed-information-to-outsider
- DisclosureInformationDisseminate:
F27 Plaintiff-disclosed-information-in-public-forum
- UseHaveSecureInformation:
F6 Plaintiff-adopted-security-measures
- HireFormEmployee:
F7 Defendant-hired-plaintiff-employee
- KnowConfidentialInformation:
F21 Defendant-knew-information-confidential

In the rules, we have not identified the entities for plaintiff or defendant, which are functional roles in a case where the role an entity plays may vary from case to case. In general, as this aspect of cases is a complex problem in itself, we have focused on the identification of key information about the factors in order to highlight candidate spans of texts.

3.10. Results

In this section, we report the results of running our gazetteer lists and JAPE rules over our corpus. The results are given as output using the GATE Annotations-in-Context tool (ANNIC). ANNIC allows one to index and search a corpus by annotation: ANNIC produces the textual span covered by the annotation, the textual spans on either side of the annotated span, the source document for each span, and the number of occurrences of the annotation in the corpus. In addition, one can search for bottom level annotations as well as combinations of them to create complex queries. We provide the results from bottom level annotations to compound factor annotations.

In Table 1, we present results for bottom level annotations, indicating the numbers of occurrences per case. Recall that these annotations are given by rule from the gazetteer lookup of lists. In turn, the gazetteer lists are intended to represent concepts given by a range of lexical items. Thus, the results are to be interpreted as the linguistic indication of the concept in the case.⁴

⁴Secure information is given in a list by phrases such as *invention agreement*, though these could have been constructed by rule.

Bottom Level	FMC	Gold.	Mid.	Tra.
Confidential	0	18	2	4
Disclosure	3	61	5	21
Disseminate	4	49	3	3
FormEmployee	2	3	5	11
Hire	0	4	7	1
Information	9	91	23	125
Know	0	7	1	17
Negotiate	10	4	4	
Outsider	6	4	5	3
SecureInformation	5	0	0	0
UseHave	37	109		72

Table 1: Bottom level annotations in cases

There are clearly relationships between these terms which merit further independent elaboration. For example, *Disseminate*, *Negotiate*, and *Outsider* relate to communications in which the parties with the information make it available to a wider audience. The properties and contexts which differentiate them will be left for future discussion.

In Table 2, we present results for pairs of annotations which are relevant to factors of three base annotations. We have used ANNIC to create searches for select pairs of annotations with 15 tokens between them without intervening sentence splitters; the results sum both orders of the pair. The results are to be interpreted as the linguistic indication of the pair of concepts in a sentence in the case.

Bottom Level	FMC	Gold.	Mid.	Tra.
Conf., Info.	0	19	1	0
Discl., Info.	0	37	7	6
Diss., Info.	2	16	3	2
Info., Out.	1	2	0	0

Table 2: Pairwise annotations in cases

The results are an overestimation in that we have not removed overlapping annotations; that is, for example, for *Information* and *Outsider*, we find both text spans “information regarding prospective customers” and “fact make information regarding prospective customers” in one case, where the latter contains the former. ANNIC does not identify the minimal span.

In Table 3, we give the factor (using the factor index) and the number of occurrences of the annotation with respect to cases in which those occurrences appear. Recall that the factors are compounds of two or more bottom level terms. Other than for F27, disclosure of information in a public forum, the results for factors are poor, though the trend is clear – the more combinations of bottom level annotations, the fewer compound annotations. In a sense, the results are surprising. Given that we have taken cases which are reported to contain the relevant factors, that we have used and broadened the terms of the factor descriptions, and that we have overlooked a range of issues that might interfere with the results, one might have expected an *overgeneration*

Factor	FMC	Gold.	Mid.	Tra.
F1	0	0	0	1
F6	1	0	0	0
F7	0	0	1	0
F10	0	0	0	0
F21	0	0	0	0
F27	0	20	5	0

Table 3: Factors in cases

of results.

To be clear, consider a range of potentially interfering issues. First, we have not taken into account reports of a fact pattern, which indicates that it holds in a case, from discussions about the concept, which do not imply the fact holds. For example, fact patterns appear as subordinate clauses under the scope of propositional attitudes or speech acts such as *believe*, *allege*, *claim*. Similarly, we have not considered fact patterns under the scope of negation *not* or terms with negative implication such as *fail* or *deny*, which again do not imply that the fact pattern holds. We have not filtered results with respect to syntactic structure. Nor have we constrained the results with respect to parties in a case. Finally, recall that our results abstract over word order.

There are a variety of ways that the results could be incrementally improved. First, we could augment the terms in the gazetteer lists given evidence from the language of the corpus and relative to the manual annotation task. In this regard, it is essential to build an accurate, richly annotated corpus manually. We should also examine the role of anaphora, ellipsis, syntactic phrases, and terms distributed across sentence boundaries. Finally, the results (in combination with the manual annotation) raise questions about the initial factor ascriptions given in (Aleven, 1997) and (Aleven, 2003); our approach does not verify the expected annotation results. One explanation is that the four cases under examination are cases on appeal rather than on first instance, which means that the facts of the cases are not under discussion but rather a point of law or interpretation. Thus, the cases in our corpus are fact pattern poor, and it remains to be verified whether the facts attributed to the cases hold or have rather been imported from the cases of first instance (e.g. by some referential mechanism). Even were this so, there is the substantive issue of whether it is correct to import such factors since the decision about the case on appeal may not rest primarily on reasoning about the factors.

At bottom, our approach emphasises an interesting question that is not highlighted in machine learning or statistical approaches – what do judges, lawyers, jurors, and law students know when they know to identify a factor in the text? Clearly, they rely on overt linguistic indicators, structure, and semantic interpretations which interact with domain knowledge. The question is important to address, for unlike other applications of information extraction where a “black box” result may be acceptable, it is highly relevant in the legal domain to provide an explicit justification and explication for a legally binding decision and to represent this in the text of the decision for subsequent reuse in case based

reasoning. Thus, despite the current results, there is strong reason to continue to investigate the phenomena. The question also touches on the representation of legal knowledge. In our annotation experiment, we limit ourselves to explicit lexical realisations of factors. However, legal knowledge would also be represented with ontologies and rules, providing intermediate level, non-lexicalized, implicit knowledge necessary for successful semantic factor annotation.

4. Manual case factor annotation

The manual annotation task performed in GATE TeamWare serves several purposes. First, it creates a gold standard, on the basis of which the predictive power of the automatically annotated low-level (e.g. bottom level) factors for high-level factor annotation can be evaluated. Second, we cannot a priori expect a full correspondence between the low-level and high-level factors. Therefore, we should also regard the manual annotation as an exploration of the interaction between the various levels of factor annotation. Thirdly, the quality of the annotations and the inter-annotator agreement can give an indication of the quality of the factors themselves. Given the rather incomplete, vaguely defined, and overlapping nature of the factors that we have available, we may expect lack of clarity amongst the annotators.



Figure 3: GATE TeamWare with low and high level factor annotations

The annotation task itself is rather complex as well. The annotators must be familiar with the semantics of the factors, and ideally agree on the exact text spans for each factor annotation. Because factors are expressed in flexible and non-predictable ways it cannot be expected that annotators agree to a high level on the exact boundaries of the linguistic text element expressing the factors under examination. In fact, this is borne out by the inter-annotator agreement results. TeamWare enables the computation of inter-annotator agreement in several ways. We have chosen

precision, recall and F1 measure. Three documents yield zero values for all, whereas one document gives 0.5 scores for precision, recall and F1. In conclusion, we see little or no agreement between the annotators of the high-level factors. In many cases, the annotators’ results are complementary rather than overlapping. We consider this an indication of the difficulty of spotting the exact lexicalisations of the complex concepts expressed by the factors. As Figure 3 shows, the low inter-annotator agreement is also partly due to overlapping, but non-identical text spans. Annotator one chose a larger text span, which includes the text span selected by Annotator two. In our opinion, this is again indicative of the highly non-trivial nature of the task.

4.1. Comparison of high-level and low-level factor annotation

In this section, we evaluate the correspondence between the factors of different levels in order to judge the predictive strength of low level factors (Low) for the selection of high level factors (High). High level factors tend to share low and compound level factors within their annotation spans. Standardly, evaluation mechanisms of precision and recall presuppose a dependency between low and high level factors which is binary – indicative or not indicative. However, in our bottom-up approach, we cannot assume that “indicativeness” is a binary notion. Rather, we postulate levels of indicativeness, where the frequency of text span enclosure is the observable measure.

Of the low level factors listed in Table 1 and the compound level factors described in Sections 3.8. and 3.9., the factors that uniquely indicate high level factors in our small corpus are in Table 4, given frequency of occurrence of the low level factor in the high level factor (Freq.) and percentage of occurrence in the corpus (Perc.).

Low	High	Freq.	Perc.
UseHaveSecureInformation	F6	1	100
SecureInformation	F6	1	20
Fair	F6	2	8.6
Appellee	F7	1	10
Defendant	F7	1	2
Hire	F7	2	16.6
Agreement	F7	2	50
Plaintiff	F10	2	3.7
Know	F21	2	8
Confidential	F21	2	4
ConfidentialInformation-TempXY	F21	1	6.6
ConfidentialInformation	F21	1	5.3

Table 4: Unique lower level indicators of high level factors

The ones with a high percentage of occurrence in the corpus, such as the compound level factor UseHaveSecureInformation and the low level factor Agreement, can be seen as strong indicators of the high level factors F6 and F7 respectively. It needs to be stressed that given the size of the corpus, we cannot make strong claims about how representative the results are of the

sub-domain we are targeting. However, a qualitative evaluation suggests that `UseHaveSecureInformation` and `SecureInformation` are both more tightly related to F6 than any other high level factor. The non-unique low level factors are shared amongst the high level factors for several reasons:

- Their presence is not indicative of the semantics of the high level factor.
- They express meaning components that are shared by the high level factors, and therefore point to vague distinctions between these high level factors. If the latter applies, an incremental refinement of the factors is needed.

5. Related work

(Brüninghaus and Ashley, 2003), (Brüninghaus and Ashley, 2005), and (Ashley and Brüninghaus, 2009) consider both knowledge representation and reasoning along with textual information processing of case factors. A system is proposed to classify cases with respect to the facts and then to predict the outcome of a case. We consider the text analysis.

(Ashley and Brüninghaus, 2009) apply NLP techniques to a squib rather than the original text of the case decision; a squib is a manually constructed summary of the case which represents the factors of the case along with factor indices; the factors are text fragments that are incorporated into a narrative. From a set of squibs, a list of positive and negative statements of each factor is manually constructed (the learning set); from the list, machine learning techniques are applied to “acquire” a classifier (a pattern) for each factor. The classifiers are applied to the test set of squibs, and each text is classified as to the factors contained within it. A *nearest-neighbour* machine learning algorithm is applied to a learning set of squibs, where the classifying pattern is compared to sentences in the test set to find sentences most similar to the classifying pattern. The success of the classification is measured against a gold standard of squibs, which have been manually classified.

Given that the results rely on the classifying pattern, several alternative representations for the learning set are considered. This means that prior to applying the machine learning algorithm, the squibs are further preprocessed using a range of NLP techniques. The three representations are:

- bag of words - the degree to which one squib is similar to another squib in terms of the lexical items in each.
- replacement - the name of an individual is replaced by their functional role in the case, e.g. IBM for plaintiff.
- “propositional patterns” - 4 pair-wise part of speech patterns such as ‘subject-verb’, ‘verb-object’, ‘verb-prepositional phrase’ and ‘verb-adjective’. A thesaurus creates alternative patterns using synonymous words within a pattern.

(Ashley and Brüninghaus, 2009) report that the F-measure, which measures the accuracy and completeness of the coverage (1 is perfect accuracy and completeness), of any of

the classification tasks is very low, for one experiment it was below 0.3. However, the reports are predominately given indirectly in terms of the impact of the representations on the results of Issue-based Prediction (IBP) of case decisions, which is a case based reasoning system.

Our approach differs from (Ashley and Brüninghaus, 2009) in several respects. First, we work with original, unstructured text rather than structured text which does not address the knowledge bottleneck at the point of identifying the factors from unstructured text. However, using structured text does have obvious advantages to unstructured text, but only to the extent that results can be extended to cover unstructured text. Second, we work with the conceptual components of the case factors (bottom level and compound annotations), and we do not apply parsing and entity extraction (e.g. party names and product information), which do not clearly provide advantages to factor identification. Indeed, since our approach generalises over the bag of words approach (incorporating a thesaurus directly to form concepts), and neither roles nor syntactic relations are relevant to further restrict output, we might have expected overgeneralised results, as we discussed earlier. Third, (Ashley and Brüninghaus, 2009) classify case squibs with respect to factors, and the factors themselves are not annotated, which implies that one cannot extract the factors per se. In our approach, factors are explicitly annotated and can be extracted. As we do not have a well-defined gold standard, we do not apply machine learning techniques, which would be premature. The source cases, squibs, and gold standard of (Ashley and Brüninghaus, 2009) are not available for public evaluation, so it is difficult to independently verify the results or contribute to the development of a factor extraction system. In contrast, we work with tools and material that promote a community development process to refine the the gazetteers and JAPE rules as well as to develop a consensus gold standard. Finally, a machine learning approach provides classifiers which may be opaque to users and which need not represent the knowledge that law students or legal professionals bring to the task of factor identification. In our approach of bottom level and compound concepts, important aspects of legal knowledge are made explicit.

6. Discussion and conclusion

We have outlined and reported an approach to annotation of legal case factors in full text decisions. It is bottom-up, starting with concepts over a range of lexical items, then constructing more complex factors from the concepts. While the results of this initial study are poor, they highlight a range of issues that can be addressed in further research – augmenting the gazetteer lists, constraining contexts under the scope of negation or propositional attitudes and speech acts, taking into account the role of ellipsis and anaphora, as well as the difference between cases of first instance and cases on appeal. We have also conducted an online, collaborative annotation task. The results indicate that further refinement of the task is required.

However, overall, we have defined a clear, well-defined, open workflow for building an annotation and extraction system for legal case factors which supports iterative refine-

ment through a collaborative process. Didactically speaking, it would be of great interest to involve law school students and legal professionals in the task of building the lists, JAPE rules, and carrying out online annotation tasks, for not only would this refine the tool, but it would encourage participants to focus on close textual analysis of the cases, which is a core capacity of every lawyer.

Given a gold standard of texts and a method to add annotated cases to the case base, we could use the extracted factors as input to a case based reasoning system such as IBP or the argument schemes of (Wyner and Bench-Capon, 2007). In addition, cases with annotated factors (in an XML compatible format) could be used for Semantic Web applications such as information extraction, querying, and reasoning with cases over the internet.

The scale of the experiment is small in terms of number of documents and of annotators. It is clear that this is just a feasibility study. A real gold standard should be created by a larger number of annotators, which will also yield statistically more reliable correspondences between lower level and higher level factors.

In future work, we will apply machine learning techniques, term extraction, ontology construction, as well as experiment with the role of syntactic structure to improve results.

7. Acknowledgements

During the writing of this paper, the first author was in part supported by the IMPACT Project (Integrated Method for Policy making using Argument modelling and Computer assisted Text analysis) FP7 Grant Agreement No. 247228.

8. References

- Vincent Aleven. 1997. *Teaching case-based argumentation through a model and examples*. Ph.D. thesis, University of Pittsburgh.
- Vincent Aleven. 2003. Using background knowledge in case-based legal reasoning: A computational model and an intelligent learning environment. *Artificial Intelligence*, 150:183–237.
- Kevin D. Ashley and Stefanie Brüningshaus. 2009. Automatically classifying case texts and predicting outcomes. *Artif. Intell. Law*, 17(2):125–165.
- Kevin Ashley. 1990. *Modelling Legal Argument: Reasoning with Cases and Hypotheticals*. Bradford Books/MIT Press, Cambridge, MA.
- Stefanie Brüningshaus and Kevin D. Ashley. 2003. Predicting the outcome of case-based legal arguments. In Giovanni Sartor, editor, *ICAIL'03: Proceedings of the 9th International Conference on Artificial Intelligence and Law*, pages 233–242, Edinburgh, United Kingdom. ACM Press: New York, NY.
- Stefanie Brüningshaus and Kevin D. Ashley. 2005. Generating legal arguments and predictions from case texts. In *ICAIL 2005*, pages 65–74, New York, NY, USA. ACM Press.
- Alison Chorley. 2007. *Reasoning with Legal Cases seen as Theory Construction*. Ph.D. thesis, University of Liverpool, Department of Computer Science, Liverpool, UK.
- Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 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)*.
- Diana E. Forsythe and Bruce G. Buchanan. 1993. Knowledge acquisition for expert systems: some pitfalls and suggestions. In *Readings in knowledge acquisition and learning: automating the construction and improvement of expert systems*, pages 117–124. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Ben Hachey and Claire Grover. 2006. Extractive summarisation of legal texts. *Artificial Intelligence and Law*, 14(4):305–345.
- Carole Hafner. 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. ACM.
- Peter Jackson, Khalid Al-Kofahi, Alex Tyrell, and Arun Vachher. 2003. Information extraction from case law and retrieval of prior cases. *Artificial Intelligence*, 150(1-2):239–290, November.
- Guiraudé Lame. 2004. Using nlp techniques to identify legal ontology components: Concepts and relations. *Artificial Intelligence and Law*, 12(4):379–396.
- K. Tamsin Maxwell, Jon Oberlander, and Victor Lavrenko. 2009. Evaluation of semantic events for legal case retrieval. In *ESAIR '09: Proceedings of the WSDM '09 Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 39–41, New York, NY, USA. ACM.
- Diana Maynard, Yaoyong Li, and Wim Peters. 2008. NLP techniques for term extraction and ontology population. In *Proceeding of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 107–127, Amsterdam, The Netherlands, The Netherlands. IOS Press.
- Marie-Francine Moens, Caroline Uyttendaele, and Jos Dumortier. 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. ACM.
- Wim Peters. 2009. Text-based legal ontology enrichment. In *Proceedings of the workshop on Legal Ontologies and AI Techniques*, Barcelona, Spain.
- Edwina L. Rissland, David B. Skalak, and M. Timur Friedman. 1996. BankXX: Supporting legal arguments through heuristic retrieval. *Artificial Intelligence and Law*, 4(1):1–71.
- Edwina L. Rissland, Kevin D. Ashley, and L. Karl Branting. 2006. Case-based reasoning and law. *The Knowledge Engineering Review*, 20:293–298.
- Adam Wyner and Trevor Bench-Capon. 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. IOS Press.
- Adam Wyner. 2008. An ontology in OWL for legal case-based reasoning. *Artificial Intelligence and Law*, 16(4):361–387, December.