CLIEL: Context-Based Information Extraction from Commercial Law Documents

Matías García-Constantino
University of Liverpool
Dept. of Computer Science
Liverpool, United Kingdom
mfgc@liverpool.ac.uk

Katie Atkinson
University of Liverpool
Dept. of Computer Science
Liverpool, United Kingdom
katie@liverpool.ac.uk

Danushka Bollegala
University of Liverpool
Dept. of Computer Science
Liverpool, United Kingdom
danushka.bollegala@liverpool.ac.uk

Karl Chapman
Riverview Law
Southwood Road
Wirral, United Kingdom
KarlChapman@riverviewlaw.com

Frans Coenen
University of Liverpool
Dept. of Computer Science
Liverpool, United Kingdom
coenen@liverpool.ac.uk

Claire Roberts
Riverview Law
Southwood Road
Wirral, United Kingdom
ClaireRoberts@riverviewlaw.com

Katy Robson
Riverview Law
Southwood Road
Wirral, United Kingdom
KatyRobson@riverviewlaw.com

ABSTRACT

The effectiveness of document Information Extraction (IE) is greatly affected by the structure and layout of the documents being considered. In the case of legal documents relating to commercial law, an additional challenge is the many different and varied formats, structures and layouts used. In this paper, we present work on a flexible and scalable IE environment, the CLIEL (Commercial Law Information Extraction based on Layout) environment, for application to commercial law documentation that allows layout rules to be derived and then utilised to support IE. The proposed CLIEL environment operates using NLP (Natural Language Processing) techniques, JAPE (Java Annotation Patterns Engine) rules and some GATE (General Architecture for Text Engineering) modules. The system is fully described and evaluated using a commercial law document corpus. The results demonstrate that considering the layout is beneficial for extracting data point instances from legal document collections.

1. INTRODUCTION

Information Extraction (IE) from legal documents is important for many reasons, including: (i) the formatted storing of the extracted data in databases, (ii) usage of the extracted data for data analysis and decision making, and (iii) input of the extracted data to some other process. The nature, amount and type of information to be extracted will depend on the business requirements: from specific information such as dates or names, to excerpts or complete sections in the documents. The challenge of IE from legal documents is the varied formats, structures and layouts used [17]; there is no standard way to represent legal documents. Format in this context refers to the different conventions that can be used to represent information in legal documents. In terms of layout, although most commercial legal documents tend to share some standard sections, the information is presented in varied ways and arrangements (two columns, position of tables, boxes containing signatures and so on); it is also not guaranteed that documents will contain the exact same sections presented in the same order. Other factors that add complexity to the IE from legal documents process are that: (i) legal documents typically contain cross-references, and (ii) different entities or parties use different conventions in terms of format, structure and layout.

From the perspective of a domain expert (lawyer), manually identifying and extracting specific information from commercial law documents (such as contracts) is a generally intuitive and straightforward process in which the effectiveness and time to complete the task will depend on the experience and personal capabilities of the expert. A domain expert is able to identify information from commercial law documents regardless of their format, structure and layout. Where there are many occurrences of some piece of required information in the text, a domain expert can effectively discriminate between them. However, given the large amount of commercial law documents that are relevant to

DOI: 10.475/123_4

a significant commercial enterprise, extracting information by hand is a time consuming process prone to human error. Automating this IE process is therefore desirable.

The central idea presented in this paper is a mechanism for annotating legal documents using XML (Extensible Markup Language) tags so as to facilitate IE of *data point types*, such as dates, legal framework, named entities and so on. We refer to specific mentions of data point types as *data point instances* in this paper. More specifically, the idea is to use a combination of methods and technologies to facilitate the desired tagging, namely: (i) NLP (Natural Language Processing), (ii) JAPE (Java Annotation Patterns Engine)[1] rules and (iii) GATE (General Architecture for Text Engineering)[2]. We refer to the resulting system as the CLIEL (Commercial Law Information Extraction based on Layout) system. The information we wish to extract is of two types: (i) sections, subsections, appendices, etc. within documents, and (ii) specific information according to business requirements (e.g. dates, names, jurisdiction). In more detail the proposed mechanism is founded on: (i) a proposed Rule-based Layout Detection (RLD) phase and (ii) a proposed Rule-based Layout Detection Tree (RLDT) data structure. The RLD phase is used to annotate, extract and parse the parts of a document into the RLDT data structure. The RLDT data structure is then used to store the identified parts and entities of a document ready for further processing. The motivation for a rule-based approach is the use of domain knowledge that can be effectively applied and extended as required. The main contribution of the work presented is the CLIEL system, a flexible and scalable IE method, aimed at the extraction of information from legal documents, regardless of format, structure or layout, by considering context. The evaluation results obtained using CLIEL demonstrated that by considering document layout data point instances from legal documents could be effectively extracted in comparison with two alternatives approaches: (i) Layout Insensitive and (ii) Majority Sense Baseline.

2. RELATED WORK

IE is a research area that has been extensively investigated as evidenced by the surveys presented in [12], [21] and [10]. Note that the first two surveys are more recent and at least ten years apart from the latter. As to be expected, IE is covered in a general way in these surveys, with the latter mentioning application areas of IE. (An IE specific system for the legal domain is presented in [19].) In the three surveys the use of rules for IE is noted. It is of particular interest that [21] mentions the Common Pattern Specification Language (CPSL) [1], which has variations such as JAPE (Java Annotation Patterns Engine) [7]; JAPE is also used with respect to the work presented in this paper.

There has been extensive research focused on different aspects of IE from legal texts. An initial distinction is between: (i) IE applied to transcripts of legal cases [6], (ii) IE directed at legislative documents [18] and (iii) IE directed at agreements of various kinds [8]. The work presented in this paper falls into the last category. In [8] a process is presented for applying IE to legal texts comprising: (i) structure parsing, (ii) handling of references, (iii) classification of sentences, and (iv) creation of model fragments. Although the work in [8] is directed at Dutch legislation texts, the work is of relevance with respect to commercial law documents. The work in [17] acknowledges the variety of legal document types and their different structures as well as the need for better ways to represent discourse patterns in legal documents. In [17] an intelligent IE system is proposed that automatically summarizes Belgian criminal cases written in Flemish.

The challenge of extracting information from legal text has lead to the use of XML formats as an effective and flexible way to represent text, where parts of documents are annotated using XML tags. With respect to legal texts, there have been some XML standards proposed in order to achieve a common XML representation, the most notable of these are: MetaLex [25] and LKIF (Legal Knowledge Interchange Format) [11]. In [3] an overview of both MetaLex and LKIF is presented. In [24], AKOMA NTOSO (Architecture for Knowledge-Oriented Management of African Normative Texts using Open Standards and Ontologies), another XML format which is intended to be use as a standard is extensively described. The use of XML to represent text offers the advantages that: (i) it allows the handling of commercial legal text in a hierarchical, organised and flexible way, and (ii) it has been widely used in academia and industry.

In addition to being used to represent and annotate legal texts, XML has been used to assist IE from legal text. An approach for automated IE, focused on the structure of documents and relationships, is presented in [16]. Here a system is described for the extraction of structure and references in Spanish normative documents digitised in various different formats (plain text, HTML, DOC, etc.); consequently there was no information loss resulting from a preliminary OCR (Optical Character Recognition) process as in the case of some other systems. Some research work has been explicitly directed at the annotation of legal text for IE. In [4] a method to automatically annotate, using XML tags, and consequently extract information from Italian legislative texts is presented. The approach presented in [26] aims to annotate and extract legal case factors in full text decisions using GATE and JAPE grammar rules as also used with respect to the work presented in this paper. The work of [9] should also be noted here because in it is presented a system for converting PDF documents into structured XML format.

The literature also reports on IE research directed at OCR digitised legal texts as in the case of the raw data considered in this paper. One example can be found in [22] where the aim is to hierarchically extract information from scanned semi-structured contracts. The work presented in [22] is similar to the research presented in this paper where the aim is IE of relevant information from legal text regardless of different layouts and client-specific features. The IE framework used in [22] is Apache UIMA[3]. While the contracts used in [22] and in the work presented in this paper have different layouts and are not client-specific, the contracts used in [22] are based on a template and the focus is on a specific type of contract (ISDA credit support annexes). The CLIEL method presented in this paper is not based on a template and is not focused on a specific type of contract.

The work of [23] also uses a hierarchical approach using Conditional Random Fields (CRF) to extract information,

[1]https://gate.ac.uk/sale/tao/splitch8.html
[2]https://gate.ac.uk/
[3]https://uima.apache.org/

in this case litigation claims and entity mentions. CRF is a probabilistic framework used to label and segment sequence data. Two hierarchical models that use CRF are presented in [23]: (i) bottom-up CRF and (ii) joint hierarchical CRF. Both models outperform the top-down cascaded approach.

NLP is used in [2] in a system based on legislative XML and ontologies called Eunomos, which is applied to manage and annotate legal text. The NLP-based system presented in [14] is called TULSI (Turin University Legal Semantic Interpreter) and is used to automatically annotate normative documents by extracting modificatory provisions, which are fragments of text that modify one or more sentences in normative text.

The approach presented in [20] combines linguistic information (lexical, syntactical and semantical) and machine learning techniques (Support Vector Machines (SVM)) to extract legal concepts and named entities. In [20] the linguistic information is used as input for the SVM algorithm, which is used to tag and extract the information of interest. An approach for extracting information from legal text using "active learning" is presented in [5]. Active learning is a type of semi-supervised machine learning in which the performance of an algorithm is optimised by the interactive input of a user. In [5], active learning is used to automatically extract licenses and to represent them in RDF (Resource Description Framework) format. The machine learning algorithms used are: (i) SVM and (ii) Multinomial Naïve Bayes.

Another approach that uses the structure and layout of a legal document to extract information is presented in [15]. This approach focuses on the concept of "elliptical lists". In the area of linguistics, "ellipses" are clauses where one or more words have been omitted. Thus the work presented in [15] automates the recognition of lists of structural items (sections, subsections, etc.) that are considered as elliptical and generate complete sentences using propositional and deontic logic.

Finally, some approaches have been implemented as systems to extract information from court decisions. In [6], an information extraction system is used to extract relevant information from decision documents of the Philippine Supreme Court written in English, particularly about criminal cases. A legal expert made recommendations on which information needed to be extracted based on its relevance and helped in the development of a template. The template was used as a guide for the information extraction. In [13], an information extraction system for Thai legal documents based on finite-state template matching is presented. The system extracts information from legal cases that is used to automatically generate summary judgments of Thai Supreme Court's verdicts. It is reported in [13] that the presented system achieves more than 90% of accuracy in terms of F-measure.

The research presented in this paper shares aspects with some of the works mentioned in this section, such as extracting information regardless of different layouts and client-specific features as well as representing the information extracted in XML format. However, CLIEL does not use templates to support in the IE as some approaches do. The current version of CLIEL is not as complex as the approaches that use a hierarchical procedure or machine learning techniques. As it is now, the CLIEL system is a flexible and scalable IE environment that has been deployed successfully on commercial contracts. Future versions could incorporate other aspects such as the use of machine learning techniques or hierarchical procedures to widen its scope and results.

3. APPLICATION DOMAIN

To act as a focus for the work presented in this paper, a set of 97 digitised commercial law documents, of a variety of formats, structure and layouts, were considered. The documents used were manually identified by a domain expert. Using the CLIEL system the objective was to process the document collection and extract data points. For evaluation purposes the documents were hand tagged so as to provide a test set. Tables 1 and 2 show the information that was annotated manually (data point types) and automatically (sections of the document), respectively. In both tables the first column indicates the type of information to be extracted, the second column presents the annotated information considered for this paper and the third column presents an example of the annotated information.

With respect to the required data points to be extracted, in commercial law, a party can have the role of customer or supplier, and in some cases both within the same document. In relation to the document test set, in all cases there were two parties involved (a customer and a supplier of a product/service); this is the reason the party and counterparty terminology is used. The phrase "governing law" refers to the legal system by which a document is governed, for example *"Law of England and Wales"*. Jurisdiction refers to the responsible authority to be referred to regarding any legal issue relating to the document. In the third column of Tables 1 and 2 examples for each case of the annotated information are shown (because of space restrictions only the index and title are shown in Table 2). Recall that the required factual data points are found in different parts of a document.

4. SYSTEM OVERVIEW

An overview of the CLIEL system is given in Figure 1. The input is a collection of legal documents in text format. The output is a collection of XML documents, each corresponding to a document in the input, where the specific required information to be extracted in each case is enclosed between XML tags. Note that while in Section 2 it is acknowledged that there are many XML representations for legal text, the simplified XML annotation scheme used in CLIEL is adequate and sufficient for processing the annotated text. The proposed automatic system comprises six main steps:

1. Application of NLP (tokeniser, gazetteer and sentence splitter) to the input text using GATE to split the text into actionable units (tokens/words and sentences) and to identify names of entities, based on predefined lists, that can be used for annotations.

2. Application of JAPE grammar rules to the text to identify and generate XML annotations of document sections.

3. Parsing of each document and translating it into an RLDT data structure where each node represents a text unit such as a title, heading or paragraph.

Type of information to be extracted	Data Point Types	Data Point Instances
Data points (Specific information according to business requirements)	Date of document	*1st December 2016*
	Name of party	*Acme Corporation*
	Name of counter-party	*Stark Industries*
	Governing law	*Law of England and Wales*
	Jurisdiction	*Courts of England and Wales*

Table 1: Manually annotated data point types from commercial law document test set.

Type of information to be extracted	Annotated Information	Example
Parts of commercial law documents to be used in Rule-based Layout Detection (RLD)	Indexes, titles and content of sections	*1. DEFINITIONS*
	Subindexes, titles and content of subsections	*1.3 A person includes a natural person, corporate or unincorporated body (whether or not having separate legal personality).*

Table 2: Automatically annotated sections from commercial law document test set.

4. Traverse the RLDT data structures in a left-first traversal to generate one XML file per document (with the text units annotated).

5. Application of JAPE grammar rules to the text to generate XML annotations for specific information.

6. Extraction of the annotated information and storage in a file or data structure.

Steps 1 to 4 are part of the RLD (Rule-based Layout Detection) phase. Note also that JAPE grammar rules are applied twice, once in Step 2 and once in Step 5. The distinction is that while in Step 2 they are used to generate XML annotations for sections in the original documents, in Step 5 they are used to generate XML annotations for specific data points, according to pre-determined business requirements, in the XML files generated from traversing the RLDT data structures in Step 4. The six steps of the CLIEL system are described below. To support the description the generic example document presented in Figure 4 will be used.

4.1 Application of NLP techniques to the input text by using GATE.

The first CLIEL step involves the application of NLP to the legal document input in order to split the text into actionable units (tokens/words and sentences) and to identify names of entities based on predefined lists that can be used for annotations. This will allow a better insight and handling of the text for the purpose of IE. GATE allows for the simple construction of pipelines with NLP modules through its own IE system called ANNIE (A Nearly-New Information Extraction System). The GATE NLP modules included in ANNIE that are of interest with respect to CLIEL are: (i) English Tokeniser, (ii) Gazetteer and (iii) Sentence Splitter. The Tokeniser splits the text so that every word is a token. In the context of NLP, a gazetteer is a list that contains known names of entities or concepts (e.g. cities, companies and surnames of people) to be identified in a text. The Splitter breaks down the text into sentences. These modules are used to support the IE process in CLIEL. The focus here is on GATE rather than ANNIE because, although the modules used correspond to the default ones that are part of ANNIE, there are several different open source NLP modules that can be integrated in an ANNIE pipeline as well as other IE systems and configurations within GATE. The output of this step is a set of annotated documents within GATE that can be selected through GATE's user interface for visualisation and, more importantly with respect to CLIEL, automatically exported as XML annotated documents. Since the generic example presented in Figure 4 does not contain names that can be annotated using a gazetteer, only tokens and sentences can be annotated, as shown in Figure 2. Note that annotated tokens and sentences are normally used within GATE to support the generation of JAPE grammar rules, as opposed to producing a set of annotated XML documents as in the case of the proposed CLIEL system. The main advantage of using CLIEL over other approaches for annotating specific named entities that are present in many parts of legal document, such as names of parties, is that the accuracy of the resulting annotation is supported by the context in which it appears in the document.

4.2 Usage of JAPE grammar rules to identify document sections.

The input to the next CLIEL stage is the GATE annotated document collection from stage 1 within GATE's GUI. As mentioned in Section 2, JAPE is a variation of CPSL, and is used to define grammar rules to annotate text documents. A JAPE grammar "contains rules which act on annotations assigned in earlier phases, in order to produce outputs of annotated entities" [7]. JAPE grammar rules are of the form: "Left Hand Side (LHS) $\rightarrow$ Right Hand Side (RHS)", where a form of regular expression of annotation pattern is defined on the LHS and a way to manipulate that annotation pattern is defined on the RHS. An example of a JAPE grammar rule to annotate the name of a person is shown in Figure 3. JAPE grammar rules, designed to identify the different sections in the document, are applied to the GATE annotated document collection in the form of "transducer" elements that are integrated in a GATE pipeline.

The output from this second CLIEL step is a set of XML annotated documents that can be selected and visualised within GATE and exported as XML files. Recall that while the output in the previous step was in the form of internal annotations within GATE, the output here is explicitly in the form of a set of XML files (one per document). Thus the annotation output from Step 1 is being extended with

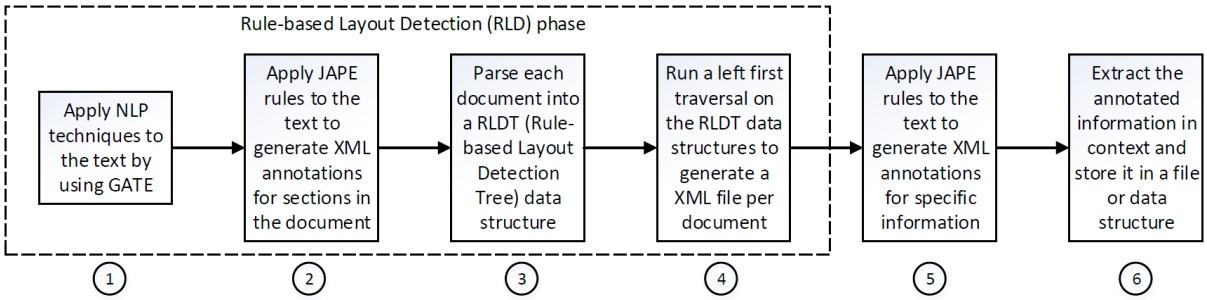

Figure 1: CLIEL workflow (box numbers are referenced in the text).

```
<Sentence><Token>1</Token><Token>.</Token><Token>FIRST</Token> <Token>TITLE</Token></Sentence>
    <Sentence><Token>(</Token><Token>a</Token><Token>)</Token><Token>Text</Token> <Token>of</Token> <Token>this</Token>
     <Token>subsection</Token><Token>.</Token></Sentence>
        <Sentence><Token>(</Token><Token>i</Token><Token>)</Token><Token>Text</Token> <Token>of</Token> <Token>this
        </Token> <Token>subsubsection</Token><Token>.</Token></Sentence>
    <Sentence><Token>(</Token><Token>b</Token><Token>)</Token><Token>Text</Token> <Token>of</Token> <Token>this</Token>
     <Token>subsection</Token><Token>.</Token></Sentence>
<Sentence><Token>2</Token><Token>.</Token><Token>SECOND</Token> <Token>TITLE</Token></Sentence>
    <Sentence><Token>(</Token><Token>a</Token><Token>)</Token><Token>Text</Token> <Token>of</Token> <Token>this</Token>
     <Token>subsection</Token><Token>.</Token></Sentence>
    <Sentence><Token>(</Token><Token>b</Token><Token>)</Token><Token>Text</Token> <Token>of</Token> <Token>this</Token>
     <Token>subsection</Token><Token>.</Token></Sentence>
        <Sentence><Token>(</Token><Token>i</Token><Token>)</Token><Token>Text</Token> <Token>of</Token> <Token>this
        </Token> <Token>subsubsection</Token><Token>.</Token></Sentence>
        <Sentence><Token>(</Token><Token>ii</Token><Token>)</Token><Token>Text</Token> <Token>of</Token> <Token>this
        </Token> <Token>subsubsection</Token><Token>.</Token></Sentence>
```

Figure 2: Sample legal text with token and sentence XML annotations.

```
Phase: firstpass
Input: Lookup Token

Rule: Name
Priority: 20
(
{Lookup.majorType == "Name"}
): name
-->
: name.Name = {rule = "Name"}
```

Figure 3: A sample JAPE grammar rule.

1. FIRST TITLE

(a) Text of this subsection.
   (i) Text of this subsubsection.
(b) Text of this subsection.

2. SECOND TITLE

(a) Text of this subsection.
(b) Text of this subsection.
   (i) Text of this subsubsection.
   (ii) Text of this subsubsection.

Figure 4: A sample legal text.

annotations of the sections in the document. The nature of the output is shown in Figure 6 with respect to the raw text given in Figure 4. From the figure it should be noted that the elements that support the identification of section boundaries within the document collection (e.g. titles and indexes) have been identified using XML tags.

4.3 RLDT Construction

In Step 3 the XML document collection from Step 2 is further processed (parsed) so that each document is translated into an RLDT structure where each node represents some text unit. Figure 5 shows the RLDT structure for our sample text from Figure 4. This is a hierarchical data structure, thus in Figure 5 the nodes at the top level (1 and 2) will hold the headings (labels) for Sections 1 and 2 respectively. The nodes at the next level in the structure will hold the headings for the corresponding subsections. At the lowest level of the RLDT structure, the actual text contained in the sub-subsections is held. The number of levels in a RLDT structure will vary depending on the layout of the document. In this way the parts of a document are now in an organised and accessible manner.

4.4 Generation of XML RLDT Document Collection.

In Step 4 the RLDT data structures generated for each document in Step 3 are traversed from left to right to generate an XML file per document. This is the second set of XML documents generated in CLIEL. The difference with respect to the set generated in Step 2 is that while in that

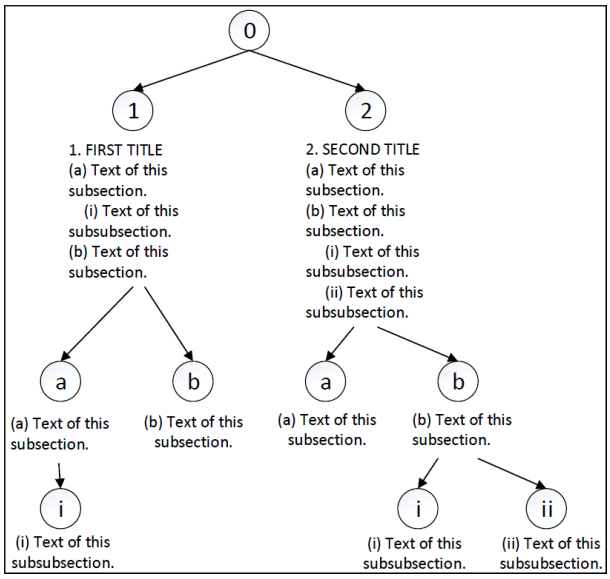

Figure 5: RLDT example.

case the XML annotations are for sections in the document, the XML annotations generated in this step are with respect to the logical representation of the RLDT data structure that is generated for each document in Step 3.

4.5 Usage of JAPE grammar rules to generate XML annotations for specific information.

In Step 5, in a similar manner to Step 2, a further set of JAPE grammar rules is applied to the XML document files from Step 4. Recall that the nature of JAPE rules applied here is for the purpose of annotating specific data point types as defined by the application domain (business requirements). The idea is to generate XML annotations for data points in the XML RLDT represented documents. The output from Step 5 is thus the XML RLDT document collection from Step 4 with further XML annotations for the, domain dependent, information of interest.

4.6 Information Extraction

The last step of the CLIEL process is the IE step, in which the XML document files generated in Step 5 containing annotations for specific information are parsed and the information extracted per document is stored in a file or in a data structure. The output is in the form of a collection of files (for example text or Comma Separated Values (CSV)) or elements of a data structure (for example tables, key-value or graphs). Recall that with respect to the initial version of CLIEL presented in this paper the aim is to extract only factual data points, which are typically located in specific sections of the document regardless of the format, layout and order of such sections. The titles of the main sections and subsections will indicate the context in which the annotated data points will be extracted. Note that in many cases multiple instances of the data points required may have been annotated; the Step 6 process discriminates between these multiple annotations.

5. EVALUATION

The evaluation of the proposed CLIEL system was carried out using the data set of 97 real commercial law documents introduced in Section 3. For the evaluation the five data point types listed in Table 3 were considered, each type may have a number of instances. Recall that each individual piece of information of interest (string) is referred to as a "data point". The objective of the evaluation was to test the proposed CLIEL IE process with respect to the quality and effectiveness of the data point extraction; the purpose was not to test the quality of the document structure identification. The factual data points of interest were manually identified with respect to each document. A subset of 20 documents was then used as a training set with which to generate JAPE rules, the remaining 77 were used for testing. The subset of 20 documents were selected by a domain expert considering how representative and varied they were with respect of the entire data set of 97 documents. To evaluate the CLIEL system, its operation was compared with two alternative approaches: (i) Layout Insensitive and (ii) Majority Sense Baseline. The Layout Insensitive approach used JAPE rules, but without the contextual (layout) information; thus without the proposed RLD phase. The Majority Sense Baseline considered the most frequent data point instance for each data point type to be the correct data point instance for that data point type in all contexts.The Majority Sense Baseline is a competitive baseline method popularly used in Word Sense Disambiguation (WSD) tasks [27]. This baseline demonstrates the importance of considering the contextual clues when extracting data point instances, and shows the level of performance we would expect if the dataset was biased towards a particular data point instance. Note that both the Layout Insensitive approach and the Majority Sense Baseline are non-context based approaches; in the first case the approach simply looked for instances of data points, in the second case the approach looked for the least ambiguous data points according to their frequency in a given document.

The layout context is given by the identified sections of the documents, which narrows the number of occurrences of data points to the sections to which they are related. In the CLIEL workflow this is defined by the JAPE rules applied after the RLD phase (see Figure 1). In all the scenarios the automatically extracted data points were tested against the manually extracted ones. The evaluation measures used were the ones typically used in the area of information extraction, namely: (i) precision, (ii) recall and (iii) the F-measure. The evaluation was conducted in the context of GATE. GATE allows three types of evaluations: (i) strict (considers partially correct annotations as incorrect with respect to the evaluation metrics), (ii) lenient (considers partially correct annotations as correct) and (iii) average (calculates the average of strict and lenient). The type of evaluation used was lenient given that partially correct annotations are usually substrings of data point strings in our training set.

The evaluation results are shown in Tables 3, 4 and 5 in terms of evaluation measures, annotations and p-values respectively. From Table 3 it can clearly be seen that the proposed CLIEL environment performed significantly better, in terms of the F-measure, than the other two approaches. The Layout Insensitive approach produced good recall values comparable with the Layout Sensitive approach (in two

```
<Index>1.</Index><Title>FIRST TITLE</Title>
    <Subindex>(a)</Subindex><Subsection>Text of this subsection.</Subsection>
        <Subsubindex>(i)</Subsubindex><Subsubsection>Text of this subsubsection.</Subsubsection>
    <Subindex>(b)</Subindex><Subsection>Text of this subsection.</Subsection>
<Index>2.</Index><Title>SECOND TITLE</Title>
    <Subindex>(a)</Subindex><Subsection>Text of this subsection.</Subsection>
    <Subindex>(b)</Subindex><Subsection>Text of this subsection.</Subsection>
        <Subsubindex>(i)</Subsubindex><Subsubsection>Text of this subsubsection.</Subsubsection>
        <Subsubindex>(ii)</Subsubindex><Subsubsection>Text of this subsubsection.</Subsubsection>
```

Figure 6: Text layout in annotated XML.

Extracted data point type	Majority Sense Baseline			Layout Insensitive			CLIEL		
	Precision	Recall	F	Precision	Recall	F	Precision	Recall	F
Date of document	0.0401	0.3207	0.0713	0.2800	**0.3325**	0.3039	**0.6473**	**0.3325**	**0.4371**
Name of party	0.1693	0.3033	0.2172	0.1763	**0.5616**	0.2667	**0.7568**	0.5541	**0.6393**
Name of counterparty	0.0843	0.2077	0.1199	0.1176	**0.6019**	0.1965	**0.6526**	0.5943	**0.6165**
Governing law	0.7388	0.5731	0.6421	0.6597	**0.8614**	0.7471	**0.9773**	**0.8614**	**0.9150**
Jurisdiction	0.9390	0.6822	0.7881	0.8104	**0.7148**	0.7591	**0.9844**	**0.7148**	**0.8271**

Table 3: CLIEL evaluation results (best results in bold font).

instances outperforming the Layout Sensitive approach).

It is also interesting to note that all three approaches were better at extracting the "Governing law" and "Jurisdiction" data points, than the other three data points. It was conjectured that this was because the "Governing law" and "Jurisdiction" data points tended to appear only in a specific section of the documents, therefore reducing the area in a document where a data point can be located, and thus simplifying the extraction process. In the case of the "Date of document", "Name of party" and "Name of counterparty" data points, the poor performance can be explained by the typically large number of occurrences of dates and names of parties in the documents. Note that most of the dates appearing in a document are usually unrelated to the actual date of the document, whilst the names of parties in a document typically appear as alias or shortened versions of the complete formal names.

The evaluation results presented in Table 3 show that the proposed CLIEL system improves the extraction of information (data points) over the other approaches by providing context in terms of document layout. In the case of "Date of document", "Name of party" and "Name of counterparty" the context was that they are usually defined or indicated in the first section of the document, the preamble section. The "Governing law" and "Jurisdiction" data points, as already noted, usually appear in a specific section of a document thus providing the context.

Table 4 presents results for the three approaches in terms of annotations considering the lenient evaluation type across the 77 documents used for testing: (i) Correct, (ii) Missing, (iii) Spurious and (iv) Partial. These results were also obtained using GATE's functionality. The annotations are defined as follows:

- *Correct*: annotations matched by the approaches with respect to the manually annotated ones.
- *Missing*: annotations from the manually annotated ones that were missed by the approaches.
- *Spurious*: annotations that were incorrectly annotated by the approaches.
- *Partial*: annotations from the manually annotated ones that were partially matched by those produced by the approaches.

An example of a partially matched annotation in the case of the Jurisdiction data point instance "Courts of England and Wales" is "Courts of England". As mentioned above, the lenient type of evaluation is being used because partial annotations typically contain useful information with respect to the data point types to be extracted. Note that the number of manual annotations is the result of the sum of correct and missing annotations. The number of manual annotations of the Major Sense Baseline differs from the other approaches because it was annotated according to the frequency of the data point instances.

In terms of annotations, the best case is to have a large number of correct annotations and a small number of spurious annotations. From Table 4, it can be seen that the number of correct annotations is almost the same for the Layout Insensitive and CLIEL approaches, however the number of spurious annotations is much higher for the Layout Insensitive approach. The Major Sense Baseline approach, in contrast, has less correct annotations and more spurious annotations than the other two approaches. For the Major Sense Baseline and Layout Insensitive approaches the number of spurious annotations for the "Date of document", "Name of party" and "Name of counterparty" data point types is particularly large. Note how the number of spurious annotations decreases from the ones produced by Major Sense Baseline and Layout Insensitive to the ones produced by CLIEL. As mentioned above, the reason why CLIEL has much less spurious annotations is the use of the context provided by the approach for each data point type. For the non-context based approaches the number of spurious annotations for the "Governing law" and "Jurisdiction" data point types is lower than the other data point types because they tend to have less occurrences within a document. In the case of CLIEL, the number of annotations for "Governing law" and "Jurisdiction" is 2 and 1 respectively because the context provided narrowed down the scope of relevant annotations. In some cases, "Name of party" and "Name of counterparty" instances were incorrectly annotated as the

Extracted data point type	Majority Sense Baseline				Layout Insensitive				CLIEL			
	Correct	Missing	Spurious	Partial	Correct	Missing	Spurious	Partial	Correct	Missing	Spurious	Partial
Date of document	37	87	964	3	40	85	110	2	40	85	23	2
Name of party	29	89	192	10	51	56	352	21	50	57	23	21
Name of counterparty	15	99	282	11	39	50	568	36	38	51	41	36
Governing law	41	42	19	13	66	14	43	16	66	14	2	16
Jurisdiction	57	29	4	4	60	26	15	4	60	26	1	4

Table 4: CLIEL annotations results.

other one probably because in both cases the name of an organisation was being extracted.

Table 5 presents the p-values resulting from two paired two-tailed t-tests for each extracted data point type: (i) CLIEL vs Majority Sense Baseline and (ii) CLIEL vs Layout Insensitive. The critical level considered was $p = 0.05$. From the t-test between CLIEL and the Majority Sense Baseline approach, statistically significant results were obtained for the following extracted data point types: (i) "Name of party", (ii) "Name of counterparty" and (iii) "Governing law". No statistically significant results were obtained for the t-test between CLIEL and the Layout Insensitive approach, the most likely reason being that the correct matches for both approaches are almost the same, what is different is that CLIEL has much less spurious annotations produced. For three data point types ("Date of document", "Governing law" and "Jurisdiction") the p-values obtained were undefined, probably non-finite quantities. Note that the JAPE rules used to extract data point instances for CLIEL and for the Layout Insensitive approach only differ in which context is considered in CLIEL, so although the p-values obtained were not significant, in practice CLIEL produces a better set of annotations.

6. CONCLUSIONS

This paper has presented a system and methodology called CLIEL (Commercial Law Information Extraction based on Layout) for extracting information from legal documents related to commercial law. CLIEL is aimed at extraction of data points from legal documents, regardless of format, structure or layout, by considering context. CLIEL uses NLP (Natural Language Processing), JAPE (Java Annotation Patterns Engine) rules and some GATE (General Architecture for Text Engineering) modules. Unlike other legal-IE systems CLIEL uses context as well as other features to extract information; context expressed in terms of document layout. More specifically the operation of CLIEL is founded on: (i) a proposed Rule-based Layout Detection (RLD) phase and (ii) a proposed Rule-based Layout Detection Tree (RLDT) data structure. The RLD phase is used to annotate, extract and parse the parts of a document into the RLDT data structure, which is then used to store the identified parts and entities of a document, in an organised and accessible way, so that it can be used for further processing. Five data point types were considered in this paper: (i) "Date of document", (ii) "Name of party", (iii) "Name of counterparty", (iv) "Governing law" and (v) "Jurisdiction".

The presented evaluation was conducted using a data set of 97 commercial law documents in which the data points of interest had been manually identified by a domain expert so as to provide a suitable benchmark data set. A subset of 20 documents was used as a training set with which to generate a set of JAPE rules. The evaluation considered three approaches: (i) Majority Sense Baseline, (ii) Layout Insensitive and (iii) CLIEL; the distinction being that CLIEL used document layout to provide context while the other two approaches did not. The evaluation measures considered were precision, recall and the F-measure. The evaluation results showed a significant improvement when using the layout sensitive strategy that was proposed with respect to the CLIEL system. Annotations results and statistical significant tests were presented to support the performance results of CLIEL with respect to the other approaches.

Note that the work presented in this paper is the first step in a larger programme of work on automated processing of commercial contracts, and as such it provides a useful foundation for future development. For future work a larger document test set will be generated with the assistance of a group of domain experts working in a commercial law environment. It could be argued that some of the data points extracted in the evaluation of CLIEL are not exclusive of commercial contracts, which demonstrates the wider applicability of CLIEL in other type of legal documents. Also for future work, non-factual and more complex information will be considered in order to improve the CLIEL method. Therefore the existing JAPE rules will be improved and extended to cover other types of factual and more complex information to be extracted. It will also be considered how to integrate and implement CLIEL as part of a more comprehensive workflow to process commercial law documentation.

7. ACKNOWLEDGMENTS

The work described in this paper was conducted as part of the "Making Sense of Legal data" Innovate UK funded Knowledge Transfer Partnership project (KTP009763).

8. ADDITIONAL AUTHORS

9. REFERENCES

[1] D. E. Appelt and B. Onyshkevych. The common pattern specification language. In *TIPSTER '98 Proceedings of a workshop*, pages 23–30, Baltimore, Maryland, U.S.A., October 1998. Association for Computational Linguistics.

[2] G. Boella, L. Humphreys, M. Martin, P. Rossi, and L. van der Torre. Eunomos, a legal document and knowledge management system to build legal services. In *International Workshop on AI Approaches to the Complexity of Legal Systems*, pages 131–146. Springer, 2011.

[3] A. Boer, R. Winkels, and F. Vitali. Proposed xml standards for law: Metalex and lkif. In *Proceedings of the 2007 conference on Legal Knowledge and Information Systems: JURIX 2007: The Twentieth Annual Conference*, pages 19–28, Leiden, The Netherlands, 2007. IOS Press.

[4] A. Bolioli, L. Dini, P. Mercatali, and F. Romano. For the automated mark-up of italian legislative texts in xml. In *Proceedings of the 2002 conference on Legal*

Extracted data point type	CLIEL vs Majority Sense Baseline	CLIEL vs Layout Insensitive
Date of document	0.634042	—
Name of party	**0.000239**	0.320484
Name of counterparty	**0.000082**	0.320484
Governing law	**0.000001**	—
Jurisdiction	0.083237	—

Table 5: P-values resulting from paired two-tailed t-tests.

Knowledge and Information Systems. JURIX 2002: The Fifteenth Annual Conference, pages 21–30, Amsterdam, The Netherlands, 2002. IOS Press.

[5] C. Cardellino, S. Villata, L. A. Alemany, and E. Cabrio. Information extraction with active learning: A case study in legal text. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 483–494. Springer, 2015.

[6] T. T. Cheng, J. L. Cua, M. D. Tan, K. G. Yao, and R. E. Roxas. Information extraction from legal documents. In *Proceedings of Eighth International Symposium on Natural Language Processing (SNLP '09)*, pages 157–162, Bangkok, Thailand, October 2009. IEEE.

[7] H. Cunningham, D. Maynard, and V. Tablan. Jape: A java annotation patterns engine. *Research memo CS-00-10, Institute for Language, Speech and Hearing (ILASH) and Department of Computer Science. University of Sheffield, UK*, pages 1–29, 2000.

[8] E. de Maat. *Making Sense of Legal Texts*. PhD thesis, Faculty of Law at the University of Amsterdam, 2012.

[9] H. Déjean and J.-L. Meunier. A system for converting pdf documents into structured xml format. In *Document Analysis Systems VII*, volume 3872 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 2006.

[10] R. Gaizauskas and Y. Wilks. Information extraction: Beyond document retrieval. *Computational Linguistics and Chinese Language Processing. Computational Linguistics Society of R.O.C.*, 3(2):17–60, August 1998.

[11] R. Hoekstra, J. Breuker, M. Di Bello, and A. Boer. The lkif core ontology of basic legal concepts. *Proceedings of LOAIT '07: II Workshop on Legal Ontologies and Artificial Intelligence Techniques*, 321:43–63, June 2007.

[12] J. Jiang. Information extraction from text. In *Mining text data*, pages 11–41. Springer, 2012.

[13] K. Kowsrihawat and P. Vateekul. An information extraction framework for legal documents: A case study of thai supreme court verdicts. In *Computer Science and Software Engineering (JCSSE), 2015 12th International Joint Conference on*, pages 275–280. IEEE, 2015.

[14] L. Lesmo, A. Mazzei, M. Palmirani, and D. P. Radicioni. Tulsi: an nlp system for extracting legal modificatory provisions. *Artificial intelligence and law*, 21(2):139–172, 2013.

[15] R. Markovich, G. Hamp, et al. Elliptical lists in legislative texts. In *Proceedings of the 15th International Conference on Artificial Intelligence and Law*, pages 192–195. ACM, 2015.

[16] M. M. Martínez, P. de la Fuente, and J.-C. Derniame. Xml as a means to support information extraction from legal documents. *Computer Systems Science & Engineering*, 18(5):263–277, 2003.

[17] M.-F. Moens, C. Uyttendaele, and J. Dumortier. Intelligent information extraction from legal texts. *Information & Communications Technology Law*, 9(1):17–26, 2000.

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

[19] E. Pietrosanti, P. Mussetto, and G. Marchignoli. Navilex: Search and navigation in a semi-automatic content acquisition legal hypertext. *Informatica e diritto, Istituto di Teoria e Tecniche dell'Informazione Giuridica (ITTIG)*, 3(2):211–234, 1994.

[20] P. Quaresma. Legal information extraction ← machine learning algorithms + linguistic information. In *Semantic Processing of Legal Texts (SPLeT-2012) Workshop Programme*, page 37, 2012.

[21] S. Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3):261–377, 2008.

[22] J. Stadermann, S. Symons, and I. Thon. Extracting hierarchical data points and tables from scanned contracts. *3rd UIMA@GSCL Workshop. International Conference of the German Society for Computational Linguistics and Language Technology*, 1038:50–57, September 2013.

[23] M. Surdeanu, R. Nallapati, and C. Manning. Legal claim identification: Information extraction with hierarchically labeled data. In *Workshop Programme*, page 22, 2010.

[24] F. Vitali and F. Zeni. Towards a country-independent data format: the akoma ntoso experience. In *Proceedings of the V legislative XML workshop*, pages 67–86. Florence, Italy: European Press Academic Publishing, 2007.

[25] R. Winkels, A. Boer, and R. Hoekstra. Metalex: An xml standard for legal documents. In *Proceedings of the XML Europe Conference*, pages 1–12, London, U.K., 2003.

[26] A. Wyner and W. Peters. Towards annotating and extracting textual legal case factors. In *Proceedings of Semantic Processing of Legal Texts (SPLeT) 2010 Workshop*, pages 36–45, 2010.

[27] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL 1995*, pages 189 – 196, 1995.