Applying GaiusT for Extracting Requirements from Legal Documents

Nicola Zeni, Luisa Mich, John Mylopoulos
University of Trento
Trento, Italy
{nicola.zeni, luisa.mich, john.mylopoulos}@unitn.it

James R. Cordy
Queen's University
Kingston, ON, Canada
cordy@cs.queensu.ca

***Abstract*—In this paper we describe the architecture and application of GaiusT, a multi-phase framework for extracting requirements from legal documents. GaiusT aims to cover the extraction process from the very first steps, pre-processing the input texts and supporting structural and semantic annotation of legal documents. The annotated information is recorded in a database facilitating both retrieval and evaluation of the results. Existing linguistic tools and resources have been applied whenever possible. A post-analysis of the ongoing implementation of the modules in the GaiusT architecture is given as a preliminary understanding of how much existing tools can be adopted and how much they had to be adapted.**

***Index Terms*—Requirements elicitation, support systems, conceptual model, annotation schema, semantic annotation**

I. USERS OF GAIUST

GaiusT is a comprehensive requirements elicitation system designed to annotate legal documents according to a general conceptual model of regulations [1], [2], [3]. The system was designed starting from Cerno [4], a semantic annotation framework whose core was coded in TXL [5].

Perspective users are requirements engineers or analysts. GaiusT could also be used by lawyers and for training junior analysts. Its architecture includes a large number of modules to covers all the steps of the elicitation process, but it does not support final users yet. Technical expertise in the programming languages used to implement the modules, knowledge of the main problems of semantic annotation, and other technical skills are required to fully exploit its functionalities. However, some modules are more stable and can be applied almost as stand alone tools. This is the case for example of the annotation module (see Fig. 1), whose interface was developed to run some experiments to evaluate the efficacy and efficiency of GaiusT [2].

From a technological point of view, the framework can be described as a development environment, which supports a variety of tasks. GaiusT has been implemented on the Microsoft Windows platform using the .NET framework (language C#). Existing tools and resources such as a PDF text extractor library, and a part-of-speech tagger have been chosen when they satisfied compatibility, modularity, integration and multilingual requirements. The current version of the system is about 50k lines of code and more than 130 MB in size.

II. THE APPROACH

The annotation process is realized in three steps: (1) *Parse*, to process the structure of the input document; (2) *Markup*, to generate semantic annotations based on annotation rules associated with concepts in the conceptual model; (3) *Mapping*, to populate a relational database with the annotations. Modules supporting these steps correspond to different layers of GaiusT architecture, which includes eight main components: (1) annotation schema generator, (2) pre-processing component, (3) TXL-rule generator, (4) document structure analyzer, (5) annotation generator, (6) database mapper, (7) evaluation component, and (8) GUI (Fig. 1). More information on these modules is given in Table II.

The high level requirements identified for GaiusT are related to features of legal documents. From a linguistic point of view, these documents are characterized by: (1) *Specialized language;* (2) *Domain dependency; (3) Structured format.*

Specialized language, because legal documents are based on a vocabulary that allows dealing with legal concepts. Examples of specialized terminology in the excerpt of the U.S. Health Insurance Portability and Accountability (HIPAA) Privacy Rule §164.512 are: "covered entity", "authorization", "permitted by this section", "agreement", "court order" or "court-ordered warrant," "subpoena", "summons issued by a judicial officer". To deal with concepts related to these terms, a conceptual model is needed. For the GaiusT framework a general model of laws has been built with the collaboration of colleagues of the Faculty of Law of the University of Trento.

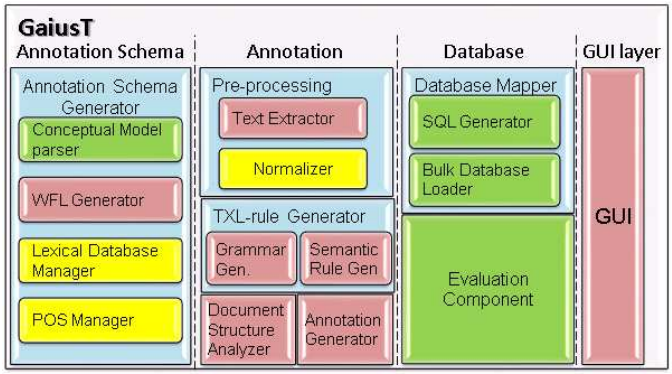

Fig. 1. Modules of GaiusT based on existing linguistic tools.

Domain specialization, because a law describes expected behaviors for a given domain, e.g., health, commerce, taxes, etc. Words and patterns characterizing legal concepts in a given domain are used in our approach to (semi)-automatically define an annotation schema. The schema specifies rules for identifying domain concepts by listing concepts identifiers with their syntactic indicators and patterns.

The HIPAA covers health issues and contains terms and expressions such as "types of wounds or other physical injuries", "blood type and rh factor". Relevant terms used in a law are often described in a glossary or in declarative parts of the law. That is the case of the HIPAA too: covered entity for example is described as "Covered entity means: (1) A health plan. (2) A health care clearinghouse. (3) A health care provider who transmits any health information in electronic form in connection with a transaction covered by this subchapter." (§160.103 Definitions). The entries in the Definitions for each section of the HIPAA have then been processed using statistical techniques (tf-itf, term frequency–inverse document frequency and ngrams) to signal candidate syntactic indicators for the concepts in the annotation schema.

Structured format, i.e., depending on the kind of document, textual content have different layout and hierarchical levels. Laws are also characterized by a nested or hyper-textual structure, as they contain internal links to other parts of the document (e.g., "as described in §164.508", "in paragraphs (f)(1) through (f)(6) of this section", "paragraph (b) or (c) of this section"), and external links, referring to other laws or regulations. For cross-references, since loops in cross-references chains are intractable (see for example HIPAA paragraph 164.528(a)(2)(i)), GaiusT recognizes and annotates cross-references, so that the annotator can take them into account and decide on a case-by-case basis. Special parsing rules catch both external and internal cross-references, and a TXL transformation renames all local labels and references to be globally unique. For example, the transformation changes list element label "(f)" of paragraph 2 of section 5 to "5.2(b)".

The structure of the HIPAA includes 12 levels (five more than the abstract model given in literature for a document), from *sentence* to *volume*. This structure is used to give information about the scope of elements of the laws: for example to specify the actors of a prescribed behavior. That is why errors in the identification of the structure issues are critical to the correct interpretation of a requirement. GaiusT annotate structural elements also to support different annotation granularity - the fragment text that refers to a concept instance, or to the amount of context that must be annotated together with the concept (e.g., lists have to be treated as a single object for a given statement). From an application point of view these choices correspond to the possibility of applying part of the conceptual model to adapt it to the goals of a given requirements extraction project. Other annotation schemas can be derived from that conceptual model also using different abstraction levels for concepts, as it happens for example in the Nomos model [6].

As regards the trade-off between recall and precision - that is, increasing recall reduces precision - for this kind of application, at any level of granularity, higher precision should be preferred. Risks associated with missing, e.g., an instance of the class actor in a right, could have serious legal consequences or worse. This means that an eager approach, in which candidate elements are reported to the analyst and manually cancelled if not appropriate, is recommended in preference to one in which the analyst has to manually look for missing elements.

Example results of items annotated by GaiusT for an excerpt of section §164.512 of HIPAA are shown in Table I. A preliminary analysis of these results confirms that structural elements are almost all correct. A rule to annotate titles of the subsections (e.g., "*Permitted disclosures: Limited information for identification and location purposes"*) should be added. Semantic annotations are more difficult to evaluate. Due to the lack of a gold standard to calculate recall and precision measures, they cannot be compared with those in [2] that are based on manual annotations of 4 experts. A preliminary analysis confirms that markups for actors, that from a linguistic point of view is a task similar to entity retrieval, are correct; other tasks, e.g. identification of actions or events, are more difficult to automatize. As an example of the complexity of the linguistic interpretation, consider that the excerpt uses "to" with different syntactic roles: "to review"; "to the applicable requirements", "is subject to paragraph (c)". GaiusT has also annotated an anti-right (see Fig. 2).

TABLE I. ANNOTATIONS FOR THE EXCERPT OF §164.512

Annotations	GaiusT
Structural	
Sections	1
Sentences	18
Titles	1
Cross-references	11
Indexes	35
Deontic	
Actors	41
Actions	10
Events	20
Resources	9
Rights	9
Anti-Rights	1

```
- <Sentence>
    <Index>(ii)</Index>
    Except as permitted by
    <CrossRef>paragraph (f)(2)(i) of this section</CrossRef>
    ,
  - <AntiRight>
      the
      <Actor>covered entity</Actor>
      may not
      <Action>disclose</Action>
      for the purposes of identification or location
    - <Constraint>
        under
        <CrossRef>paragraph (f)(2) of this section</CrossRef>
      </Constraint>
      any
      <Information>protected health information</Information>
      related to the
      <Actor>individual</Actor>
      's DNA or DNA analysis,
    </AntiRight>
    dental records, or typing, samples or analysis of body fluids or tissue
  </Sentence>
.
```

Fig. 2. Excerpt of GaiusT annotation for §164.512 of the HIPA

III. DISCUSSION

A. Research Challenges Addressed

Our approach defines a multi-phase framework to semi-automatically identify deontic concepts in legal documents. Through a defined conceptual model it is possible to build, in a semi-automatic way, an annotation schema to be used in the annotation of the documents. GaiusT also supports the identification of the hierarchical structure of legal documents and the identification of complex patterns. All this information is useful for extracting requirements from legal documents.

B. Benefits that Could not Be Demonstrated with the Excerpt

Structured format. Only a subset of the structured format of the HIPAA is clear from the small example extract from §164.512 (it represents only about a 0.7% of the law). Also, part of the section ((paragraphs (a) to (e)) has been omitted, so that it is not possible to correctly interpret, e.g., the reference to "paragraph (b)(1)(ii) or (c)(1)(i) of this section".

Multilinguality, given that a software system could have to satisfy requirements extracted from laws in different languages, GaiusT has been applied to an Italian law to investigate problems related to language issues. Language diversity are treated at three levels: structural (to deal with, e.g., characters or word splitting rules of Italian), syntactic (the annotation schema must include indicators in Italian) and semantic.

C. Limitations and Unaddressed Research Questions

Common knowledge must also be dealt with properly by annotation tools, because, for example, "care", "plan", etc. are not defined in the law texts. GaiusT uses common knowledge when the annotation schema is defined. As a research question, semantics (and pragmatics) required to fully understanding a law have the same complexity of a general-purpose natural language processing systems.

Multi-alphabet, related to the multilingual issue, more and more frequent with the globalization of the economy (e.g., e-commerce website had to complain to laws in different countries). Software companies developing information systems for banks working in different countries, as e.g. in China or Arabia do have to deal with ideograms or the Arabian alphabet. That implies problems with the direction in which documents have to be read. To the best of our knowledge there are no research on this issue as concern requirements extraction from legal documents.

D. Evaluation and Validation

To evaluate the efficacy and efficiency of GaiusT two experiments have been conducted. The first case study is based on the US Health Insurance Portability and Accountability Act (HIPAA, in English), while the second analyzes the Italian accessibility law, legge Stanca, for information technology instruments (both in Italian and in English). Results of the experiments are described in [3]. Evaluation was accomplished by comparison against human performance (gold standard). The manual evaluation of the quality of automatic results was carried out for the following sections contained in the analyzed parts: §164.520: Notice of privacy practices for protected health information; §164.522: Rights to request privacy protection for protected health information; §164.524: Access of individuals to protected health information; and §164.526: Amendment of protected health information. On the basis of the experimental study, GaiusT was able to identify legal requirements with high precision (from 93 to 100%). Good recall rates were also demonstrated for most concepts (70 to 100%), apart from anti-obligation (33%), where the tool's performance must be improved. Productivity was also evaluated, to test the usefulness of the tool for non-experts in regulatory text who may have to analyze such documents to generate requirements specifications for a software system. An empirical validation of the proposed tool against inexperienced requirements engineers has been carried out. Results showed that novices who performed full annotation of a given section of the HIPAA reported a recall of 52% with a comparable accuracy and an high error rate, while novices who simply improved GaiusT annotations reported a much higher rate of precision and recall (95% and 90% respectively) with an accuracy of 90% [2]. As regards the results for the Italian Stanca Act, The system demonstrates good precision for all of the concepts, ranging from the lowest 67% (for Actor) to the highest rate 100% for Right, Anti-Obligation and Constraint. Recall was not as high as precision, showing the lowest rates of 33% for Right.

E. Tool Support

The extraction process is supported by several components that start from input documents, annotate and map the results into a database for late analysis. At a meta-level, a post-analysis of the implementation of the modules in the GaiusT architecture is given as a preliminary understanding of how much existing tools can be adopted and how much they had to be adapted. The large number and variety of linguistic tools and resources available online represents a challenge per-se. Their applications to support requirements extraction from legal documents add some more problems. Natural language processing is an area in which a lot of problems are far from being solved and the large variety of existing tools poses a challenging question: to what extent is it possible to use them for the development of systems to help requirements analysts? And then, which activities of the requirements extraction process could be supported? How much adaptation is needed? Is the adaptation and integration of available modules preferable to build them from scratch? A simplistic view of the problem, could lead one to think that a search and adapt approach would work for almost all the functionalities of a requirements extraction system. A throughout analysis of the problem, in terms of required functionalities and performances, types of input, and requested output, allow highlighting its complexity and suggest the need of a decision making approach. In Fig. 1 and in Table II the modules of GaiusT are colored using the semaphore metaphor: in green (light gray) modules developed using existing linguistic tools in an effective way, in yellow (gray) modules that are based on existing tools but requiring heavy adaptation, and in red (dark gray) modules developed from scratch. Table II also gives information on the applicability of existing linguistic tools in terms of availability, scalability and portability.

TABLE II. LINGUISTIC TOOLS FOR THE MODULES OF GAIUST

	Components and modules	Input	Output	Tools	Evaluation (Y=yes, N=no)
1 ANNOTATION SCHEMA GENERATOR					
	Conceptual Model Support	graphical conceptual model	XMI file	UML modeling tool	Availability: ok; scalability: more Y than N Portability: java based open source software
	Conceptual Model Parser	XMI, RDF, or OWL file	preliminary annotation schema (structured text file)	Parser	Availability, scalability: Y Portability: java based open source software
	WFL (word frequent list) Generator	plain text documents	DB with one table for each file; an inverse WFL for each input file; a summary inverse WFL	Parser, stemmer, tf/idf, database mapper	Availability, scalability, portability: more likely N than Y
	Lexical Database Manager	preliminary annotation schema	table with list of syntactic indicators for the concepts	WordNet, Thesaurus, Gazeteer	Availability, scalability, portability: more likely N than Y
	POS Manager	plain text document	list of words with the probabilistic syntactic role	POS	Availability, scalability, portability: more likely N than Y Language: available only for some languages
2 PRE-PROCESSING COMPONENT					
	Text Extractor	doc, rtf, pdf, Web pages (HTML, Php, Asp, etc), XML	plain text file	Text extractor	Availability, scalability: more likely N than Y
	Normalizer	plain extracted text	cleaned plain text	Parser	Availability, scalability, portability: more likely Y than N
3 TXL GENERATOR COMPONENT					
	Grammar Generator	preliminary annotation schema	grammar file	Parser, regular expression generator	Availability, scalability: more likely N than Y
	Semantic Rule Generator	preliminary annotation schema	rule file	Parser, regular expression generator	Availability, scalability, portability: more likely N than Y
4	**Document Structure Analyzer**	plain text	annotated document with structure elements	Parser, regular expression generator	Availability: more likely Y than N Scalability: more likely N than Y
5	**Annotation Generator**	annotation schema, grammar file, rules, TXL program, plain text document	XML annotated document	Parser	Availability: more likely Y than N Scalability, portability: more likely N than Y
6	**DATABASE MAPPER**			Scripiting	Availability: more likely Y than N Scalability, portability: Y
	SQL Generator	XML document	set of SQL statements	Scripiting	
	Bulk Database Loader	set of SQL statement	relational database (Oracle, MS SQL Server or MySQL)	Scripiting	Availability, scalability, portability: Y
7	**Evaluation Component**	XML document(s), the manually annotated version	database with as many tables as input XML document	Scripiting	Availability, scalability, portability: Y
8	**GUI**			Client - Web	Availability, scalability: more likely N than Y Portability: more likely N than Y

REFERENCES

[1] N. Zeni, "The Cerno framework for semantic annotation: extensions and applications", PhD thesis, University of Trento, Trento, Italy, 2008.

[2] N. Zeni, N. Kiyavitskaya, J. R. Cordy, L. Mich, and J. Mylopoulos, "GaiusT: supporting the extraction of rights and obligations for regulatory compliance", unpublished.

[3] N. Kiyavitskaya, N. Zeni, T. D. Breaux, A. I. Antón, J. R. Cordy, L. Mich, and J. Mylopoulos, "Automating the extraction of rights and obligations for regulatory compliance", Proc. 27th inter. conf. on Conceptual Modeling, Springer, doi: 10.1007/978-3-540-87877-3_13.

[4] N. Kiyavitskaya, N. Zeni, Cordy, J.R., Mich, L., and J. Mylopoulos, "Cerno: lightweight tool support for semantic annotation of textual documents", Data&Knowledge Engineering, vol. 68, n. 12, pp. 1470–1492, 2009, doi:10.1016/j.datak.2009.07.012.

[5] J. R. Cordy, "TXL – a language for programming language tools and applications", Proc. 4th Int. Work. on Language Descriptions, Tools and Applications, Electronic Notes in Theoretical Computer Science, vol. 110, 2004, pp. 3–31.

[6] A. Siena, I. Jureta, S. Ingolfo, A. Susi, A. Perini, and J. Mylopoulos: "Capturing variability of law with Nómos 2", Proc. ER 2012, Proc. 31st inter. conf. on Conceptual Modeling, Springer, pp 383-396, doi: 10.1007/978-3-642-34002-4_30..