Sistema Modular de Apoyo para el Estudio, Interpretación y Análisis de Leyes Venezolanas LANIE: Law Analysis through Information Extraction

Alexy José Sánchez Henríquez

Postgrado en Modelado y Simulación de Sistemas,
Universidad de Los Andes, Mérida 5101, Venezuela.
Fecha: 3 de octubre de 2005.

Este anteproyecto plantea crear un sistema modular que sirva de apoyo para el estudio, interpretación y análisis de las leyes venezolanas, basándose en técnicas de procesamiento del lenguaje natural (PLN), inteligencia artificial (IA), y Extracción de información (EI). El sistema que proponemos implementar estará en capacidad, por una parte, de extraer definiciones y tópicos de los artículos de las leyes que integran el conjunto de leyes venezolanas (CLV); y, por otra, de utilizar la información extraída para descubrir de manera asistida implicaciones lógicas y contradicciones entre los artículos o definiciones presentes en una de las leyes de entre el CLV. Además, la extracción de los tópicos de los artículos permitirá la creación de resúmenes seudo constructivos de las leyes. Con todo lo anterior, se pretende proporcionar un sistema de ayuda importante para la interpretación, estudio y análisis de las leyes por parte de sus usuarios.

El presente anteproyecto de tesis plantea el diseño e implementación de LANIE (*Law Análisis through Information Extraction*), un sistema modular de apoyo para el estudio, interpretación y análisis del conjunto de leyes venezolanas (CLV), basado en técnicas de la *Inteligencia Artificial* (IA), de una de sus subdisciplinas llamada *Procesamiento del Lenguaje Natural* (PLN) – y específicamente del ámbito de la *Extracción de Información* (EI).

El sistema que proponemos implementar estará en capacidad, por una parte, de extraer definiciones y tópicos de los artículos de las leyes que integran el conjunto de leyes venezolanas (CLV) y, por otra parte, de utilizar la información extraída para descubrir de manera asistida implicaciones lógicas y contradicciones entre los artículos o definiciones presentes en una de las leyes del CLV. Además, la extracción de los tópicos de los artículos permitirá la creación de resúmenes seudoconstructivos de las leyes. Con todo lo anterior, se pretende proporcionar un sistema automatizado de ayuda para la interpretación, estudio y análisis de leyes.

II. Planteamiento del problema. Antecedentes y Objetivos.

La presente propuesta surge a partir de reflexiones hechas en la realización de un sistema de extracción de definiciones para el CLV denominado *ExDef*, diseñado e implementado como proyecto del curso *Procesamiento y Modelos de Recuperación*

de Información dictado en este postgrado como tópico especial[1].

La justificación del diseño e implementación del sistema que aquí se propone es inherente a la naturaleza misma de los sistemas de Recuperación y Extracción de información: La ingente cantidad de información que se genera y divulga aunada a la necesidad manifiesta de parte de la Sociedad de la Información[2] y de los profesionales para obtener información de forma selectiva y eficaz. En el ámbito jurídico la situación no es diferente. Venezuela, que es el caso que nos atañe directamente, cuenta en la actualidad con más de 600 leyes en vigencia. Además, el país está atravesando en estos momentos por un período de reforma política, originando así, entre otras dos consecuencias. En primer lugar, la creación de leyes para regir ámbitos que no habían sido tomados en cuenta anteriormente; en segundo lugar, la reforma de leyes que se consideran obsoletas o incongruentes con el nuevo orden legal que se desea establecer. Por otra parte, estas mismas reformas políticas que se vienen suscitando en el país exigen la revisión continua de las leyes, tanto la de vieja data como las más recientes, no sólo por parte de los legisladores sino también de los profesionales del derecho y del público general, que tiene entre uno de sus derechos, el acceso a la información, y entre uno de sus deberes, el conocimiento de las leyes. Son, entonces, los argumentos antes esgrimidos los que nos han motivado a pensar en la implementación de un sistema que coadyuve al análisis, estudio e interpretación de las leyes venezolanas.

Con el fin de comprender mejor el concepto de lo que se plantea hacer, presentamos algunas definiciones claves. Realizar un ***análisis*** consiste en distinguir y separar las partes de un todo hasta llegar a conocer sus principios o elementos (DRAE, 2005). El ***PLN*** (también conocida por sus siglas en inglés, NLP – ***Natural Language Processing***), es como lo define WIKIPEDIA, una subdisciplina de la ***IA***, y también de la ***Lingüística Computacional*** (LC), que estudia

[1] Cf. Sánchez, 2005.

[2] El termino *sociedad de la información* es de por sí polisémico, pero aquí se entenderá como *Conglomerado humano cuyas acciones de supervivencia y desarrollo esté basado predominantemente en un intensivo uso, distribución, almacenamiento y creación de recursos de información y conocimientos mediatizados por las nuevas tecnologías de información y comunicación.* (tomado de *Portal de la Información en Cuba,s/f*).

los problemas inherentes al procesamiento y manipulación de lenguajes naturales y que tiene entre sus principales tareas de trabajo la *Recuperación y Extracción de la información* (Wikipedia,s/f). La *IA*, que data desde finales de los años cincuenta, puede ser definida como aquella *inteligencia* exhibida por artefactos creados por humanos (es decir, artificial) y que a menudo se aplica hipotéticamente a los computadores. Este término es, además, polisémico, puesto que también se usa para referirse al campo de la investigación científica que intenta acercarse a la creación de tales sistemas (idem). Por su parte, la *LC* es entendida como un campo multidisciplinar de la lingüística y la informática que utiliza la informática para estudiar y tratar el lenguaje humano, y que para lograrlo, intenta modelar de forma lógica el *lenguaje natural* desde un punto de vista computacional. En el modelado participan lingüistas, informáticos especializados en inteligencia artificial, psicólogos cognoscitivos y expertos en lógica, entre otros (idem). Entre sus áreas de estudio se encuentran el diseño de analizadores (*parser* en inglés) para lenguajes naturales y el diseño de etiquetadotes o lematizadores (*tagger* en inglés) como el *POS tagger* (idem). Se utiliza el término *lenguaje natural* para referirse principalmente al lenguaje humano. Además, el término se utiliza también en contraposición a los lenguajes formales, como por ejemplo los lenguajes de programación, el lenguaje matemático o lógico; o *lenguajes artificiales* como el Esperanto que han sido creados por el hombre con un fin explícito.

A partir de las definiciones anteriormente presentadas aclaramos que nuestro estudio está enmarcado en los ámbitos de la IA y de la LC con el propósito de modelar y crear un sistema modular para distinguir y separar las partes de una ley del CLV por medio de la EI con el fin de analizarlas. Ahora bien, ¿Porque proponemos un sistema modular para la consecución de nuestro objetivo? Los sistemas modulares tienen por finalidad la coordinación de módulos para resolver problemas complejos. Como se verá más adelante en el planteamiento de los objetivos generales y específicos, los módulos que hemos planteado modelar y crear darán al

sistema global un comportamiento complejo simulando un comportamiento inteligente. Un modelado así se hace necesario porque el lenguaje no es un cuerpo inerte; por el contrario, lo que le permite a un lenguaje mantenerse como medio de expresión de una comunidad es su flexibilidad y capacidad de reinventarse y adaptarse a las necesidades de un mundo cambiante, en el que constantemente surgen nuevos conceptos y con ellos la necesidad de crear nuevos términos y construcciones para representarlos. Esto hace necesario modelarlo de manera también dinámica con módulos que presenten comportamientos propios de los humanos – como reflexión, planificación y toma de decisiones autónomas – para poder así contar con la flexibilidad necesaria para adaptarse a las nuevas formas morfológicas y sintácticas. Por otra parte, si bien es cierto que los textos correspondientes al dominio jurídico presentan características generales comunes, también es cierto que los contenidos en ellos presentes son distintos; incluso, la temática misma por la cual fueron creados impone la elección y utilización de unidades lingüísticas simples, sintagmáticas y fraseológicas que pueden variar mucho de un texto a otro.

A continuación se presentan los antecedentes que respaldan esta propuesta.

Antecedentes

El campo de la Lingüística Computacional y de la Inteligencia Artificial son en la actualidad campos de mucho auge en los que especialistas de distintas áreas muestran mucho interés. Ya hemos mencionado la importancia que tiene hoy en día la implementación de programas y sistemas que ayuden al lector a hacerse de información pertinente de forma eficaz. Concretamente, en la disciplina del Procesamiento del Lenguaje Natural, son muchos los trabajos publicados en congresos internacionales, como por ejemplo SPIRE (*String Processing and Information Retrieval*) anteriormente llamada WSP (*WorkShop on String Processing*) que se realiza anualmente, o los congresos realizados por la Sociedad Española para el Procesamiento de Lenguaje Natural (SEPLN), donde se resume la actividad de PLN para la

lengua española[3]. Existen también revistas indizadas y arbitradas en formato papel o digital que publican sobre el tema, como la revista *Lecture Notes on Computer Science* (LNCS) que tiene una serie denominada *Lecture Notes on Artificial Intelligence* (LNAI), y que publica trabajos en PLN y EI (*Lecture Notes on Computer Science, s/f*). Existen además distintos grupos de investigación como el Grupo de Investigación en Procesamiento de Lenguaje Natural y Sistemas de Información (GPLSI) de la Universidad de Alicante (GPLSI, s/f).

En relación con el ámbito de la Extracción de Información, anualmente se realiza la MUC (Conferencias sobre sistemas para la Compresión de Mensajes), la cual cuenta cada vez con más y más participantes provenientes de distintas universidades y centros de investigación a nivel mundial[4]. En lo que atañe a sistemas o programas para Extracción de Información a partir de leyes escritas en español, desconocemos la existencia en la actualidad de programas o sistemas de este tipo, con la excepción del sistema *Exdef*, el cual fue concebido como una propuesta para extracción de definiciones en leyes venezolanas a partir de un corpus conformado por la *Ley Orgánica del Trabajo*. *Exdef* es considerado entonces como un prototipo para el sistema que aquí proponemos en lo que se refiere a los módulos de análisis morfosintáctico y semanticopragmático, junto con el módulo de extracción de definiciones, puesto que el sistema presenta limitaciones, entre las cuales se incluye el hecho de que fue diseñado a partir de un corpus muy limitado (50.628 ocurrencias). Otra limitación del estudio es el hecho de que *Exdef* trabaja sólo con definiciones introducidas por el marcador de definición *entenderse* en tiempo presente. No obstante, *Exdef* cuenta con características importantes que sirven de punto de partida para el sistema que aquí proponemos y que muestran su viabilidad. Entre ellas podemos contar el hecho de que *Exdef* no cuenta con diccionarios, sino con un repertorio de conjunto de elementos finitos de la lengua (v.g. artículos, preposiciones, etc.). *Exdef* hace uso de estos elementos finitos para

[3] Las referencias a estos congresos están disponibles en la página de la SEPLN (http://www.sepln.org)

[4] Las MUC fueron creadas en 1987. Anualmente se realizan conferencias en las que se entregan a los participantes textos de entrenamiento para el diseño de sus sistemas.

realizar un análisis morfosintáctico parcial del texto y definir, en conjunto con los marcadores de definición, los límites del término a ser definido y su definición. Los límites son establecidos por medio de una serie de reglas lógicas establecidas a partir de reglas sintácticas del español.

Otro antecedente importante de mencionar es el *resumidor de textos automático* realizado por Hilda Yelitza Contreras (Contreras, 2002). Este resumidor es el resultado de su tesis de maestría en la que propone el diseño e implementación de un resumidor aplicando técnicas simbólicas basadas en una gramática de estilo que modela reglas propuestas por Williams (Williams, 1990). Este programa es capaz de *obtener desde los tópicos de las oraciones de cada párrafo y reconocer elementos sintáctico-estructurales de cohesión y coherencia textual, hasta los tópicos más importantes del párrafo* (Contreras, 2002: p. 2). La metodología propuesta por Contreras resulta muy apropiada para la obtención de tópicos de los artículos que serán utilizados para la realización de resúmenes seudoconstructivos de las leyes y para resolver problemas de cohesión y coherencia en las leyes, lo que permitirá extraer información completa cuando haya referencias anafóricas.

En concreto, para la consecución de nuestro sistema deberemos cumplir con los objetivos que se plantean a continuación.

Objetivos

General

Diseñar e implementar un sistema modular de apoyo para el análisis de leyes venezolanas.

Específicos

.- *Modelar y crear un módulo que realice el análisis morfosintáctico parcial del español utilizado en las leyes venezolanas*. Se pretente que este módulo *constate* ciertas categorías gramaticales (e.g. sustantivo, verbo, adjetivo) e identifique la función de los principales componentes de un enunciado escrito (sujeto, predicado con sus distintos complementos).

.- *Modelar y crear un módulo que segmente la ley tanto en su macroestructura*

(e.g. Nombre de la ley, entidades que decretan la ley, objeto de la ley, número y nombre de los títulos que conforman la ley, número y nombre de los capítulos que constituyen cada título expresados en la ley, número de artículos que conforman cada capítulo y tópicos de los artículos) como en su *microestructura* (parágrafos, literales y referencias).

.- *Modelar y crear un módulo semanticopragmático que extraiga definiciones contenidas en las leyes*. Este módulo contará con una base de conocimiento de operadores metalingüísticos explícitos (OMEs) (Rodríguez, 1999, 2005) presentes en el dominio del sistema – e.g. *entenderse, definirse*, etc.

.- *Modelar y crear un módulo que realice resúmenes seudoconstructivos para las leyes*. Este módulo contará con una base de conocimiento (conformada por la información que le suministre el modulo segmentador – e.g. las distintas partes que componen la ley bajo análisis).

.- *Modelar y crear un módulo que realice comparaciones entre artículos o definiciones de una ley y establezca relaciones de implicación lógica o contradicción*. Este módulo contará con una base de conocimiento (información que le suministren otros módulos – e.g. artículos, definiciones – y contará también con conocimiento de cómo identificar implicaciones lógicas y contradicciones – e.g. reglas de inferencia), la capacidad de tomar decisiones – e.g. decidir mostrar al usuario si existe alguna relación de implicación lógica o contradicción con alguna definición o artículo que se haya revisado.

.- *Diseñar e implementar la Interfaz Gráfica de Usuario para la interacción hombre- maquina en el análisis de leyes venezolanas*.

III. Metodología y Plan de Trabajo

El proceso o metodología de desarrollo de software puede ser definido como un marco de trabajo de las tareas que se requieren para construir software de alta calidad; esto es, software fiable y que funcione eficazmente (Pressman, 1998). Existen distintos modelos de procesos o metodologías de desarrollo de software propuestos por la Ingeniería del

Software, definida como la aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software (IEEE, 1993). La visión del sistema que planteamos diseñar e implementar es la de un sistema modular con licencia GNU escalable que pueda ser modificado y extendido para satisfacer nuevas y diferentes necesidades, con componentes que puedan ser reutilizados para la creación de otros sistemas de PLN. Por tal razón, se hace imprescindible que su diseño e implementación se realice siguiendo normas de calidad y que se obtengan productos (modelos, documentación y productos terminados) que satisfagan la visión antes expuesta. En tal sentido, nos proponemos seguir uno de los modelos de procesos o métodos de desarrollo de software propuestos por la ingeniería de software (Pressman, 1998): El *modelo iterativo incremental basado en casos de uso*.

A continuación daremos una breve explicación de este modelo y de cómo resulta adecuado para el desarrollo de nuestro sistema. El modelo iterativo e incremental se fundamenta en el perfeccionamiento gradual del sistema visto como un todo. En este modelo, se aplican secuencias lineales de análisis, diseño, generación de código y prueba para los distintos componentes que integran el sistema por cada iteración. No excluye la creación de prototipos que se puedan ir perfeccionando en cada iteración. Los componentes se van desarrollando en orden de complejidad, siendo por lo general el núcleo del programa el que primero se desarrolla. Posteriormente, se generan nuevos componentes apoyados en el núcleo o en otro componente inmediato anterior. Creemos que este modelo es adecuado para el desarrollo de nuestro sistema, el cual contará con un módulo que realizará el análisis morfosintáctico superficial y la identificación de las partes de la oración. Este módulo se convertirá en el núcleo que permitirá posteriormente la creación de otro módulo que permita extraer la macroestrutura u orden retórico de las leyes. Posteriormente y basado en los productos de los dos modulos anteriores, se diseñará un módulo que extraiga las definiciones de las leyes y un último

módulo que realice comparaciones entre artículos de una misma ley para establecer implicaciones lógicas o contradicciones. El segundo módulo también permitirá el modelado de un módulo que se encargue de crear resúmenes seudoconstructivos. Además, por su misma naturaleza iterativa, este modelo nos permitirá crear modelos básicos basándonos primeramente en un número reducido de leyes y validarlos para luego extenderlos gradualmente con el fin de abarcar el mayor número de leyes posible.

Por otra parte, el modelo iterativo e incremental resultará útil porque sostenemos que los distintos componentes que se desarrollen tendrán la autonomía necesaria para ser reutilizados independientemente o en otras aplicaciones. A continuación presentamos la Figura 1, en la que se muestra un diagrama del modelo incremental e iterativo.

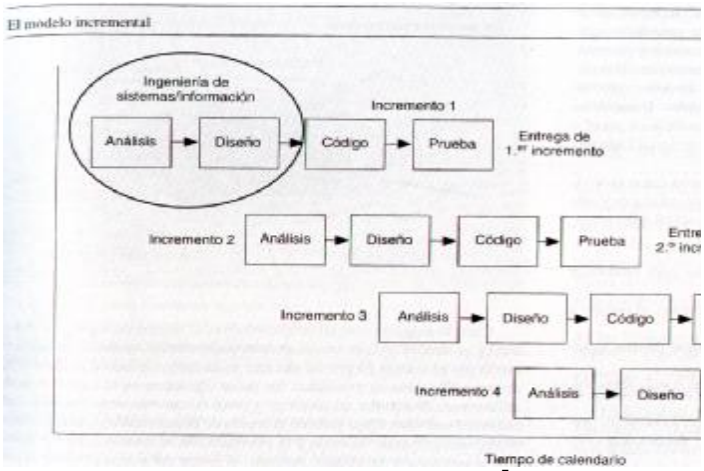

Figura 1. Modelo iterativo incremental[5]

A continuación mostramos las tareas que serán realizadas en cada iteración.

Análisis: Se especifican los requisitos, comportamiento, rendimiento e interconexión del componente a ser desarrollado.

Diseño: A partir del análisis del componente se realiza su modelado utilizando el lenguaje de modelado de software UML (*Lenguaje de Modelado Unificado*).

Generación de código: Implementación del componente siguiendo también un modelo iterativo incremental; esto es, incorporando nuevas y funciones más complejas en cada ciclo.

Pruebas: Corridas piloto que permitan evaluar la validez de los algoritmos y que el componente hace lo que se supone debe hacer. También se sigue aquí un modelo iterativo, esto es, se comienzan las pruebas con textos sencillos y luego se incrementa el grado de complejidad de los textos a procesar.

Mantenimiento: En esta fase se podrán realizar tres tipos de modificaciones: (1) correctivas que revisarán y corregirán posibles errores detectados en la implementación del componente; (2) de adaptación, que modificarán el componente para incorporar nuevas y funciones más complejas; (3) De prevención, que modificarán la interfaz del componente para

[5] Tomado de Pressman (1998).

acoplarse mejor a los requisitos del componente futuro.

Plan de Trabajo

A continuación mostramos la programación de actividades para la realización de las tareas descritas en la sección de la metodología, la cual presupone la inversión de un tiempo no mayor a 24 semanas.

Semanas 1 y 2: Documentación.

1) Estudio del lenguaje de modelado UML
2) Estudio de mecanismos de inferencia

Semanas 3 y 4: Especificación de requisitos y diseño de modelo del módulo para la realización del análisis morfosintáctico

Semanas 5 y 6: implementación y pruebas del modelo

Semanas 7: Especificación de requisitos y diseño de modelo del módulo para la extracción de distintas partes que conforman las leyes.

Semanas 8 y 9: implementación y pruebas del modelo.

Semana 10: Especificación de requisitos y diseño de modelo del módulo para la extracción de definiciones contenidas en las leyes.

Semanas 11 a 13: implementación y pruebas del modelo.

Semana 14: Especificación de requisitos y diseño de modelo del módulo para la realización de resúmenes seudoconstructivos.

Semanas 15 y 16: implementación y pruebas del modelo.

Semana 17: Especificación de requisitos y diseño de modelo del módulo para la comparación de artículos o definiciones contenidas en una ley.

Semanas 18 y 19: implementación y pruebas del modelo.

Semanas 20 y 21: Diseño e implementación de la interfaz gráfica de usuario.

Semana 22: implementación y pruebas del sistema como un todo.

Semanas 23 y 24: Elaboración del informe correspondiente para el proyecto.

El programa de actividades que aquí presentamos es indicativo y puede sufrir modificaciones en función del ritmo de trabajo real y de las problemáticas de la investigación que puedan surgir en el transcurso de su realización.

IV. Referencias

Contreras, H. (2002). Una técnica para la extracción automática de resúmenes basada en una gramática de estilos. Tesis de Maestría, Maestría en Computación, Universidad de Los Andes. Mérida, Venezuela.

DRAE (2005). Diccionario de la Real Academia Española (23a Ed.). Disponible en http://buscon.rae.es/diccionario/drae.htm

GPLSI (s/f). Grupo de investigación en Procesamiento del Lenguaje Natural y Sistemas de Información. Disponible en http://gplsi.dlsi.ua.es/ (Fecha de acceso: septiembre de 2005)

IEEE (1993). Estandards Collection: Software Engineering. IEEE Standard 610.12-1990.

Lecture Notes on Computer Science (s/f). Disponible en http://www.springer.de/comp/lns (Fecha de acceso: Septiembre, 2005).

Portal de la Información en Cuba (s/f).

Disponible en http://www.uh.cu/facultades/fcom/portal/indice.htm (Fecha de acceso: Septiembre, 2005)

Pressman, R. (1998). Ingeniería del software: un enfoque práctico. Madrid: McGraw-Hill

Rodríguez, C. (1999) *Operaciones metalingüísticas explícitas en textos especializados.* Barcelona: Institut Universitari de Lingüística Aplicada, Universitat Pompeu Fabra [trabajo de investigación, Doctorado en Lingüística Aplicada].

Rodríguez, C. (2005) *Metalinguistic information extraction from specialized texts to enrich computational lexicons*. Tesis doctoral, Facultat de Traducció i Interpretació, Universitat Pompeu Fabra, Barcelona. (URL: http://www.tdx.cesca.es/TDX-0228105-114717/)

Sánchez, A., y Melva Márquez. (2005). Hacia un sistema de extracción de definiciones en textos jurídicos. Trabajo inédito realizado como proyecto del curso de tópico especial *Procesamiento y Modelos de Recuperación de Información*. Postgrado en Modelado y Simulación de Sistemas. Mérida, Venezuela.

SPLN (2005). Disponible en http://www.sepln.org

Wikipedia(s/f). Disponible en http://es.wikipedia.org/wiki/Portada

Williams, J. (1990). Style: Toward clarity and grace. Chicago and London: The University of Chicago Press.