Método para la representación semi-automática de modelos conceptuales desde documentos de negocio escritos en lenguaje natural en español

A method for semi-automatic conceptual models representation from business documents written in natural language in spanish

Diego Alejandro Marín-Alvarez^{1*}

Bell Manrique-Losada¹

Juan Bernardo Quintero¹

Recibido 15 de julio de 2020, aceptado 1 de septiembre de 2020

Received: July 15, 2020 Accepted: September 1, 2020

RESUMEN

Actualmente la industria de desarrollo de software presenta grandes retos para el procesamiento de información de negocio, particularmente aquella contenida en documentos textuales. En el proceso de educación de requisitos de software una fuente potencial de información relevante son los documentos de negocio, pues a partir de éstos se puede facilitar la comprensión de conocimiento de un dominio, así como conocer la evolución de un producto. A pesar de su utilidad, los ingenieros de requisitos no siempre la usan para su labor por los tiempos y costos que implica. En el presente trabajo se aborda esta problemática y se reconoce por medio de una revisión sistemática, la potencialidad de usar técnicas de Procesamiento de Lenguaje Natural (NLP por sus siglas en inglés) para extraer información textual relevante de documentos de negocio, y la utilidad de representarla en modelos conceptuales. A partir de esto, se propone un método semi-automático de extracción de información desde documentos de negocio escritos en lenguaje natural en español y su representación en un modelo conceptual. El método se soporta en un marco metodológico de referencia para proyectos de Analítica de Texto, se fundamenta en técnicas de NLP, y se representa la salida en un diagrama de clases. El método fue evaluado mediante un caso de estudio con analistas de software en Medellín-Colombia, tomando como entrada documentos de resolución de telecomunicaciones. La evaluación permite concluir que el modelo es una satisfactoria aproximación a solucionar el problema planteado mejorando el tiempo de procesamiento y manteniendo un nivel de interpretación similar al proceso manual.

Palabras clave: Procesamiento de lenguaje natural, POS Tagging, modelo conceptual, educación de requisitos.

ABSTRACT

Currently, the software development industry presents challenges for processing business information, specifically that contained in textual documents. In the process of software requirements elicitation, a potential source of relevant information is business documents, since they can facilitate the knowledge understanding about a domain, as well as know the evolution of a product. Despite its usefulness, requirements engineers do not always use it for their work because of time and costs involved. In this paper this problem is addressed and it is recognized through a systematic literature review, the potentiality of using Natural Language Processing (NLP) techniques to extract relevant textual information from business documents, and the utility of its representation in conceptual models. Starting from this, a semi-automatic method of extracting information from business documents written in natural language in Spanish and its representation in a conceptual model is proposed. The method is supported in a reference

¹ Universidad de Medellín. Facultad de Ingeniería. Medellín, Colombia.

E-mail: dialmarin17@gmail.com; bmanrique@udem.edu.co; juquintero@udem.edu.co

* Autor de correspondencia: dialmarin17@gmail.com

methodological framework for Text Analytics projects, is based on NLP techniques, and the output is represented in a class diagram. The method was evaluated through a case study with software analysts in Medellin-Colombia, taking as input telecommunications resolution documents. The evaluation allows us to conclude that the model is a satisfactory approach to solving the problem, and some lines of work are identified to generalize a solution.

Keywords: Natural language processing, POS Tagging, conceptual model, requirements elicitation.

INTRODUCCIÓN

En el marco de desarrollo de software la adquisición y representación de conocimiento se realiza mediante modelos o diagramas construidos por un profesional que debe identificar las fuentes de información que le sean útiles y de allí extraer los datos que requiere para documentar dichos modelos, todo esto basado en sus habilidades para obtener información de diferentes fuentes, así como de su experiencia [1].

Durante el proceso de educación de requisitos se ejecutan entrevistas, sin embargo, los stakeholders pueden estar obviando información importante [2]; como complemento, se puede obtener información desde otras fuentes, como documentos de negocio [3]. Algunos de los documentos de negocio contribuyen al conocimiento del dominio [4], son escritos en lenguaje natural y se reconocen como documentos de procesos, documentos de requisitos, especificación de diseño, escenarios de casos de uso, entre otros. Obtener y analizar la información contenida en estos documentos es importante para comprender la evolución y el dominio de un producto software [4].

Los principales inconvenientes presentados al procesar documentación en una organización radican en la extracción misma de la información, ya que esta no se encuentra estructurada bajo marcos previamente definidos, lo cual conlleva a procesos de extracción de información repetitivos que dan como resultado requisitos que no cumplen las necesidades planteadas [5]. Con lo anterior, cualquier mecanismo que permita crear modelos que representen conocimiento de un dominio son útiles, además en casos donde un analista de negocio sea nuevo en un proyecto o área funcional, estos modelos ayudan a extraer y analizar los requisitos asociados a la construcción de un producto [6].

Gracias al incremento en tamaño y complejidad de los sistemas y de la información misma, hay

una gran demanda por enfoques inteligentes que aporten en el proceso de Ingeniería de requisitos [7]. Además, se cuenta con gran cantidad de información textual, pero los ingenieros de requisitos no tienen suficiente tiempo para leerla y analizarla; de ahí que las tecnologías para la extracción automática, capaces de presentar la información de manera concisa, están adquiriendo gran importancia [8]. Así, puede considerarse útil el uso de Procesamiento de Lenguaje Natural (NLP por sus siglas en inglés) dado que tiene como entrada textos [9] y utiliza diferentes técnicas con el propósito de lograr un procesamiento de lenguaje similar al humano para una gama de tareas o aplicaciones [10], lo que podría permitir agilizar el tiempo que toma leer y entender la información contenida en los textos.

En la literatura actual se evidencia que en los últimos años han surgido estudios que proponen diferentes enfoques para la extracción y representación del conocimiento, los cuales ejecutan procesos de NLP para procesar información escrita en lenguaje natural; la literatura también muestra una preferencia muy marcada a procesar textos escritos en idioma inglés, además se destaca que dentro de los métodos de NLP usados para la extracción de la información hay una clara tendencia a la implementación del método *Part of Speech Tagging* o *PoS Tagging*. El conocimiento extraído es representado en su mayoría mediante ontologías, las cuales requieren conocimientos previos para ser interpretadas, o mediante modelos UML, los cuales son bastante populares gracias a su baja curva de aprendizaje y sus notaciones visuales [11].

Con base en las tendencias antes descritas, el presente trabajo adopta *PoS Tagging* como método de extracción y modelos UML para la representación del conocimiento, planteando que aplicar técnicas de NLP a documentos de negocio escritos en idioma español permite apoyar el proceso de extracción de información y generar de manera semi-automática un modelo conceptual que represente conocimiento de dominio de un proceso de

negocio. Esta extracción y representación se realiza de una manera más rápida que el proceso completamente manual realizado por un ingeniero de requisitos, asegurando que se mantenga la comprensión de la información del negocio contenida en los textos.

En este trabajo se propone un método semi-automático de representación de modelos conceptuales, a partir de la extracción de información desde documentos de negocio escritos en lenguaje natural en idioma español. Para su implementación se adaptan algunas herramientas que permiten realizar pre-procesamiento de documentos de negocio para extraer elementos relevantes del texto y definir reglas de mapeo entre elementos identificados en el pre-procesamiento y elementos de un modelo conceptual, y además, se desarrolla una herramienta que permite transformar los elementos extraídos en elementos de un modelo conceptual. El método propuesto fue validado aplicándolo a un caso de estudio con un grupo de analistas de software en Medellín-Colombia, tomando como entrada documentos de resolución de telecomunicaciones, buscando identificar el tiempo de procesamiento y el nivel de interpretación de la información. Los resultados de la evaluación son satisfactorios y permiten concluir que el modelo es una buena aproximación a solucionar el problema planteado, y se identifican algunas líneas de trabajo para lograr una solución general.

El resto del artículo se organiza de la siguiente manera: en la primera sección se presenta el contexto inicial de la investigación describiendo el problema planteado y la hipótesis. En la segunda sección se muestra un análisis del estado del arte mostrando tendencias para solucionar el problema planteado. En la siguiente sección se presenta el método propuesto para solucionar el problema y se describe en detalle el proceso de implementación y validación. En la sección de hallazgos se muestran los resultados de la experimentación con su respectivo análisis, para finalmente dar las conclusiones y líneas de trabajo futuro derivado de esta investigación.

CONTEXTO DE LA INVESTIGACIÓN

Referentes conceptuales

Part-of-speech tagging (POS tagging)

Es un proceso en el cual cada palabra en un texto se asigna a su categoría morfosintáctica apropiada

(por ejemplo, sustantivo-singular, verbo-pasado, adjetivo, pronombre, y similares) [12]. Por lo tanto, proporciona información sobre la morfología (estructura de las palabras) y la sintaxis (estructura de las oraciones). Este proceso de desambiguación se determina tanto por las restricciones del léxico (¿cuáles son las posibles categorías para una palabra?), como por las restricciones del contexto en el que se produce la palabra (¿cuál de las categorías posibles es la correcta en este contexto?).

Modelado Conceptual

El modelado conceptual consiste en describir la semántica de las aplicaciones de software en un alto nivel de abstracción [13]. Los modeladores conceptuales pueden describir (1) modelos de estructura en términos de entidades, relaciones y restricciones; (2) modelos de comportamiento o funcionales en términos de estados, transiciones entre estados y acciones realizadas en estados y transiciones; o (3) interacciones e interfaces de usuario en términos de mensajes enviados y recibidos, y apariencia [13]. En su uso típico, los diagramas de modelos conceptuales son abstracciones de alto nivel que permiten a los clientes y analistas entenderse entre sí y permitir a los analistas comunicarse con éxito con los programadores de aplicaciones.

DESCRIPCIÓN DEL PROBLEMA

Durante la fase de ingeniería de requisitos en el ciclo de vida de desarrollo de software se evidencian grandes retos dado que se debe obtener información de diferentes fuentes [15]; obtener esta información en forma parcial o no interpretarla en forma adecuada, puede derivar en baja calidad de los requisitos [15-16]. El inconveniente principal radica en que las organizaciones cuentan con gran cantidad de información escrita en lenguaje natural, pero las personas no tienen suficiente tiempo para leerla y analizarla dado que los textos en lenguaje natural pueden ser muy extensos [17].

Figura 1. Ejemplo de modelo conceptual [14].

La utilización de técnicas de NLP para el procesamiento de este tipo de documentos puede ayudar a mejorar el tiempo que toma a los ingenieros de requisitos en leer y comprender la información allí contenida sin que se comprometa el nivel de interpretación de la misma.

En este trabajo se plantea como hipótesis que aplicar técnicas de NLP a documentos de negocio escritos en idioma español permite extraer información y generar un modelo conceptual de una manera más rápida que el proceso manual realizado por un ingeniero de requisitos. Este proceso asegura que se mantenga la comprensión de la información del negocio contenida en los textos.

REVISIÓN DE LITERATURA

Se realizó una revisión de la literatura basada en la metodología de revisión sistemática presentada por Beltrán [18], la cual propone hacer una definición del problema, especificar los criterios de inclusión y exclusión de los estudios, luego formular un plan de búsqueda de la literatura, pasando al registro de los datos y evaluación de la calidad de los estudios seleccionados, para finalmente hacer Interpretación y presentación de los resultados.

Partiendo del problema planteado previamente, se incluyen artículos que cumplen con los siguientes criterios de inclusión:

- Trabajo primario basados en NLP publicados a partir del año 2012.
- Deben presentar algún modelo de representación de conocimiento, preferiblemente en ingeniería de software, aunque no se descartan enfoques en otras áreas de investigación.
- Los artículos seleccionados deben utilizar fuentes de información organizacionales.

El plan de búsqueda se enfocó en las bases de datos de ACM, IEEE Xplore, Science Direct - Elsevier, Springer Link y SEPLN; y en algunos casos, se tomaron artículos referenciados por trabajos primarios o secundarios resultado de las búsquedas ejecutadas.

A continuación, se presentan los resultados obtenidos de la revisión, organizados en dos categorías, la primera muestra trabajos que presentan un modelo de representación, en la segunda trabajos que

procesan documentos de negocio sin entregar un modelo específico de representación.

Generación de modelos de representación de conocimiento desde información de negocio escrita en lenguaje natural

Diamantopoulos, Roth y Symeonidis [19] proponen procesar requisitos funcionales escritos en inglés generando un corpus lingüístico representado mediante una ontología.

Iqbal y Bajwa [20] proponen la generación de diagramas de actividades UML a partir de requisitos transformados manualmente de lenguaje natural a notación SBVR (*Semantic of Business Vocabulary and Rules*) mediante PoS (*Part of Speech*).

En el trabajo de Bhala y otros [21] se propone la generación automática de un modelo conceptual desde requisitos escritos en lenguaje natural en idioma inglés. Este trabajo es una buena aproximación a la solución del problema planteado, sin embargo, requeriría intervención para su aplicación al idioma español.

El aporte de Annervaz, Kaulgud, Sengupta y Savagaonkar [6] se orienta hacia procesar requisitos escritos en lenguaje natural y un modelo de dominio en Excel escrito en forma manual. El mayor problema de este trabajo es que requiere intervención manual para usar el modelo presentado.

Lo más interesante de los trabajos anteriores es que permitirían procesar documentos de requisitos de software existente para entender cómo funciona dicho software. De Araujo y otros [22] generan una propuesta de extracción de información desde texto en lenguaje natural que combina conocimiento de negocio con conocimiento lingüístico.

El aporte de Roychoudhury [23] consiste en la verificación automática de documentos de dominios de negocio, enfocado en: 1) crear conocimiento a través de una ontología y 2) proveer un framework de automatización para extraer definición formal del texto escrito en lenguaje natural. En este caso se requiere intervención manual para pre-transformar del documento. Xiao [17] propone la extracción de reglas de ACP (*Access Control Policy*) a partir de documentos en lenguaje natural aplicando análisis sintáctico, para representarlas al final en XACML (*eXtensible Access Control Markup Language*).

Lee [24] propone un framework para generar restricciones en OCL (*Object Constraint Language*) desde el texto en lenguaje natural entregado por el usuario, aplicando *POS Tagging*. Los dos aportes anteriores permitirían tomar documentación de reglas de negocio para ser transformadas en un lenguaje etiquetado, el principal inconveniente es que requiere que el texto se escriba en forma controlada, lo que requiere pre-procesar los documentos.

Pinquie [25] propone la aplicación de técnicas de NLP a requisitos escritos en lenguaje natural en idioma inglés, para generar PPBR (*Parametric Property-Based Requirements*). Este trabajo en particular se acerca mucho a solucionar el problema planteado en esta propuesta, sin embargo, está aplicado al idioma inglés y el modelo de representación entregado requiere conocimientos adicionales para ser interpretado por los analistas, en detrimento del tiempo de procesamiento.

El trabajo publicado por Sawant y otros [26] consiste en forzar la estructura en casos de uso potencialmente sin estructura extrayendo el modelo de oraciones comúnmente encontradas en casos de uso de la industria. Fernandes, Furquim y Lopes [27] presentan un método para extraer términos específicos de un dominio específico.

Como se evidencia en la Tabla 1, el modelo de representación dominante son las ontologías y/o lenguajes etiquetados, con siete trabajos. Tres de los trabajos representan la información extraída en modelos UML, sin embargo, ninguno de estos se procesa desde el idioma español.

Si bien el uso de ontologías marca una tendencia, los modelos UML son bastante populares gracias a su baja curva de aprendizaje y sus notaciones visuales [11]. Tomando en cuenta lo anterior y basados en que, como se ve también en la Tabla 1, es posible representar información de negocio en modelos UML, la propuesta acá presentada adopta este enfoque, de modo que se genere un modelo de dominio representado en un modelo conceptual.

Aplicación de técnicas de NLP para extracción de conocimiento desde información de negocio escrita en lenguaje natural

Los trabajos que se presentan a continuación, si bien no generan un modelo de representación como tal, procesan información de texto en lenguaje natural desde documentos de negocio o se procesa información en español, por lo que, en cierta medida, estarían aportando a la solución del problema planteado en el presente trabajo.

El aporte de Escartín [28] consiste en la comparación de diferentes aplicaciones *PoS tagger* comparando los resultados arrojados por cada uno para proponer el que mejor resultado arroje como un estándar. Si bien este trabajo no presenta un modelo de representación específico, es el único que aplica NLP directamente a textos escritos en idioma español.

Gaw [29] propone el procesamiento de reglas de negocio expresadas en lenguaje natural controlado, validando si los métodos generados en código fuente corresponden a una regla de negocio. Lo más relevante de este trabajo es que procesa documentos

Tabla 1. Idioma y modelo de representación presentado.

Referencia	Idioma del texto procesado	Técnica de extracción principal	Modelo presentado	Año de publicación
[22]	Portugués	Parsing	Ontología	2013
[27]	Portugués	Herramienta de mapeo	Ontología	2013
[6]	Inglés	Parsing	Ontología	2013
[23]	Inglés	Machine Learning	Ontología	2016
[19]	Inglés	PoS Tagging	Ontología	2017
[21]	Inglés	PoS Tagging	Modelo UML	2014
[20]	Inglés	PoS Tagging	Modelo UML	2016
[24]	Inglés	PoS Tagging	Modelo UML	2014
[26]	Inglés	PoS Tagging	Modelo propio	2014
[17]	Inglés	PoS Tagging	Lenguaje etiquetado	2012
[25]	Inglés	PoS Tagging	Lenguaje etiquetado	2015

de reglas de negocio, sin embargo, requiere que estas se escriban en un lenguaje natural controlado, por lo que no cualquier documento de reglas puede aplicarse.

Aa, Leopold y Reijers [30] presentan la detección automática de inconsistencias entre descripciones textuales de procesos y modelos de procesos, combinando análisis lingüístico, medidas de similitud semántica y orden de relaciones.

Miranda [31] un nuevo modelo para la generación de resúmenes de un documento basado en grafos conceptuales aplicando el analizador *Parser* de Stanford, su fuente de información son textos escritos en idioma inglés.

Movshovitz y Cohen [32] proponen análisis código fuente para predecir comentarios de documentos que contienen mezcla de código fuente y texto, aplicando modelos estadísticos y análisis semántico a código fuente comentado para extraer información textual en inglés. El trabajo de Naeem [33] consiste en la generación de *Queries OLAP* a partir de texto escrito en lenguaje natural controlado.

Keith, Fuentes y Meneses [34] presentan un método que aplica *POS Tagging* a textos en español para obtener la estructura de una oración y luego, con el uso de diccionarios, determinar la orientación semántica del texto. Miranda y Guzmán [35] proponen extraer información desde la web para luego identificar los elementos que corresponden a un modelo de acción para tareas de Planeación automática. Los planes se recuperan desde la web y se procesan usando herramientas de pos tagging y stemming de NLP, para guardar todas las acciones extraídas en una ontología. Jara, Chacón y Zelaya [36] proponen un trabajo en el que se toman diagnósticos médicos escritos en el lenguaje natural-controlado en español UMLS® (por sus siglas en inglés para *Unified Medical Language System*®).

Se usa machine learning para clasificar y codificar cada diagnóstico evaluado.

A continuación, se resumen los hallazgos con respecto al idioma procesado y la técnica de extracción de información, teniendo en cuenta los trabajos vistos previamente en las dos categorías:

Como se muestra en la Figura 2, quince de las propuestas se aplican al idioma inglés, dos se aplican a portugués y tres al español. Teniendo en cuenta que los trabajos aplicados a español no entregan un modelo de representación, como aporte a la solución del problema planteado en la presente investigación se buscará su generación, a partir de textos en español.

De la Figura 3 se puede inferir que el método principal utilizado para el procesamiento de lenguaje es *PoS Tagging* con once trabajos, los demás utilizan *Parsing*, herramienta de mapeo o *machine learning*. Esto muestra una tendencia importante de lo que se puede aplicar para la solución del problema, además, los trabajos que procesan texto en español usan *PoS Tagging*.

MÉTODO DE EXTRACCIÓN Y REPRESENTACIÓN

Consideraciones de la Propuesta

A partir de la hipótesis planteada, la solución propuesta se centra en un método de extracción de información desde documentos de negocio

Figura 2. Cantidad de trabajos por idioma procesado.

Figura 3. Cantidad de trabajos por técnica de NLP.

escritos en lenguaje natural en idioma español y su representación en un modelo conceptual, partiendo de las siguientes consideraciones:

- Teniendo en cuenta la tendencia evidenciada en la revisión de literatura, donde la técnica base de NLP a implementar más apropiada es el *Part Of Speech Tagging (POS Tagging)*, el método aplica esta técnica para extraer elementos específicos del texto como sustantivos, verbos, y adverbios, los cuales pueden ser mapeados a un modelo de representación.
- La representación de la información se plantea mediante el uso de modelos conceptuales, teniendo en cuenta que estos muestran conceptos significativos en un dominio [14] y permiten ser representados en diagramas UML. Estos diagramas son bastante populares gracias a su baja curva de aprendizaje y sus notaciones visuales [11].

- El método se sustenta en varias herramientas de software que permiten aplicar la técnica definida y generar el modelo de representación, apoyadas en herramientas existentes.

Marco metodológico de referencia

La propuesta de solución planteada en este trabajo se orienta bajo el marco metodológico propuesto por Manrique y otros [37] para desarrollar proyectos de analítica de texto desde documentos de requisitos. La propuesta metodológica se especifica y representa en la Figura 4.

De esta especificación se perciben tres tipos de artefactos que podrían ser los insumos: documento de negocio, documento de requisitos e historias de usuario. Los artefactos resultantes son el documento de contexto del proyecto, el corpus, el modelo, el análisis de resultados y el informe final del proyecto. En cuanto a las herramientas a usar se identifican

Fuente: Manrique y otros [37].

Figura 4. Propuesta metodológica para desarrollo de proyectos de analítica de texto.

tres tipos fundamentales, los cuales para este tipo de proyectos son automáticas:

- Preparador (pre-procesador): se encarga de tomar los documentos que conforman el corpus y unificarlos en uno solo. En este caso se usan o adaptan herramientas de software que permiten compilar documentos y/o entrenar algoritmos de NLP o *Text Mining*, usando el corpus inicial y transformándolo en uno propio para cada caso, según lo requiera el compilador.
- Compilador (procesador): se encarga de hacer el análisis sintáctico, gramatical, semántico, entre otros; y extraer los conceptos más relevantes del negocio. Da como resultado una especificación en lenguaje controlado (*i.e.* ontología). En éste se configuran herramientas que realizan procesamiento de lenguaje natural o *Text Mining*, como *Freeling*, *Stanford NLP* y otras, las cuales permiten ejecutar procesos como *POS Tagging* y *Lemmatization*.
- Modelador: se encarga de tomar los conceptos de la especificación de salida y representarlos en un modelo más estructurado. Normalmente corresponde a herramientas construidas en cada proyecto, las cuales permiten transformar la especificación en diagramas que pueden ser vistos gráficamente desde herramientas como *Enterprise Architect* o *Ed Graph Editor*.

Los roles implicados son: (1) director del proyecto, quien se encarga del ‘documento de contexto del proyecto’ para definir los objetivos y el alcance; (2) documentador, quien se encarga de las actividades centrales que permiten generar el modelo a partir de los documentos, y (3) implementador, quien se encarga de verificar que el modelo se ajuste a las necesidades de los desarrollos en los que se utiliza y hace el respectivo seguimiento del uso de los modelos resultantes.

Método de extracción propuesto

El marco de referencia es tomado como base en razón a que permite organizar las actividades propias de proyectos que responden a la línea de investigación seguida en el presente trabajo, además es un marco presentado en el contexto local, donde, de acuerdo a lo determinado en el estado del arte, hay poco desarrollo de proyectos de NLP, lo que permite que el caso de estudio

acá presentado sirva como validación del marco mismo.

Teniendo en cuenta que se trata de una propuesta experimental, es necesario establecer que no se cuenta con roles claramente definidos, en este caso existe un único rol que ejecuta todo el proyecto; además, al ser experimental, sólo se implementa hasta la fase 5 de Evaluación donde se toman los datos necesarios para el análisis. El método propuesto está planteado de forma que se preparan las herramientas de software para procesar un documento de negocio cada vez, mientras la propuesta metodológica base plantea el procesamiento de muchos documentos a la vez. En la Figura 5 se muestra el modelo general del método propuesto y a continuación, se describe cada fase.

Fase 1. Contextualización

Esta fase está basada en la definición del problema y la definición de la hipótesis; el objetivo consiste en el análisis automático de documentos de negocio para la extracción de información relevante y su representación en un modelo conceptual. En el caso de estudio abordado en este trabajo en particular se toma como artefacto output lo contenido en la sección CONTEXTO DE LA INVESTIGACIÓN.

Fase 2. Recolección

En esta fase se realiza la compilación de “documentos de negocio”, se determina los documentos que sirven como base para el modelo, desde los cuales se obtienen términos relevantes para el dominio en particular, y se identifican los documentos que se usarán en la fase de evaluación.

El caso de estudio en particular, y tomando como input el contexto mencionado en la fase anterior, determinó el uso de documentos de resolución emitidos por la Comisión de Regulación de Comunicaciones CRC de Colombia, los cuales son de dominio público, estos son considerados como el corpus de documentos, conformado por 7 documentos de resolución.

Fase 3. Preparación

En esta se ejecutan dos labores fundamentales, la primera aplica para la preparación de las herramientas de software para procesar cada documento, así:

- Determinar las herramientas más idóneas para la implementación.

Figura 5. Método propuesto para generar un modelo conceptual a partir de un documento de negocio.

- A partir de los documentos que conforman el corpus inicial, se toman los términos de negocio más importantes y se les da un peso de negocio de 1 a 10, donde 1 es menos importante y 10 es más importante. Estos términos servirán posteriormente para priorizar elementos en el modelo conceptual resultante. Esta clasificación es conocida como *Information content (IC)*, métrica que denota la importancia de un término en un corpus o en un dominio [38], enmarcada en la técnica de procesamiento de texto *Bag Of Words*. El resultado de esto queda en archivo plano, que luego es usado por la herramienta.

Se toman los sustantivos como términos de negocio dado que para el método propuesto estos hacen parte fundamental de la primera regla de mapeo, lo cual se explica en la fase 4. Una vez obtenidos los términos se hizo una definición de peso de negocio (*Information content (IC)*) para cada término, labor ejecutada con el apoyo de un analista experto en el dominio.

Una vez finalizado lo anterior, se procede a procesar los documentos elegidos para la prueba del documento (sea resolución o fragmento de resolución) ejecutando las actividades 1 y 2 (para cada documento) de la herramienta de software, explicados en la sección

Implementación de herramienta para automatizar el método. En este caso, como artefacto de output se tiene la definición de pesos de negocio para el dominio específico y el texto depurado del documento que se está procesando.

Fase 4. Modelamiento

En esta fase se procesa el texto y se obtiene un modelo conceptual para cada documento, así:

- Generar POS Tags:** Mediante herramientas de clasificación como *Freeling*, se obtiene una clasificación de partes de oración de los términos del texto extraídos, esto con una salida en XML.
- Procesar POS Tags:** Luego, se analiza el XML resultante de modo que se transforme a elementos de un modelo conceptual, específicamente conceptos y relaciones. Esta identificación se hace de acuerdo a lo identificado en la literatura, donde se especifica que:
 - Las clases de un modelo conceptual se pueden obtener usando nombres y frases nominales [14].
 - Un dominio consiste en conceptos y relaciones, así, en un modelo conceptual los conceptos se identifican como entidades. El nombre de una entidad debe ser un sustantivo [39].

- Cuando se crea un modelo de dominio, una buena fuente de clases de dominio incluye requisitos de alto nivel. Es útil analizar estos requisitos, extrayendo los sustantivos y las frases nominales, luego, puede refinarlos para crear el modelo de dominio inicial [40].

Con lo anterior se define entonces la primera regla de transformación de elementos extraídos del texto en elementos de un modelo conceptual, así, un elemento identificado como sustantivo se convierte en un concepto del modelo. En términos del resultado de *Freeling* se identifican con el valor “noun” en la propiedad “pos” de cada término en el XML resultante.

La cantidad de conceptos identificados en cada documento puede ser muy alta, esto genera un modelo confuso que se ve saturado por conceptos y dificulta su lectura y comprensión; por lo que es necesario no mostrar en el modelo final todos los conceptos, sino los más importantes, por esto se deben priorizar. Para lo anterior se siguió la técnica de procesamiento de texto *Bag of Words* con *Information Content (IC)* [38]. La técnica permite obtener una lista de palabras con la cantidad de veces que se repite en el texto, lo cual se conoce como *Term Frequency (TF)*, este se multiplica por el *Information content (IC)* de cada término (explicado en la fase 3 preparación), obteniendo una medida que para este trabajo llamaremos TFIC, y se ordena de mayor a menor TFIC.

La cantidad final de conceptos a mostrar puede variar, así, se obtiene un número óptimo de conceptos a mostrar en el modelo en función del número de páginas. El usuario puede cambiar este número sugerido por la herramienta.

Las demás reglas de mapeo sirven para identificar las relaciones entre conceptos, para lo cual se definen unos patrones de *POS Tags*. Dado que estos patrones pueden variar, se parametrizan en un archivo plano junto con las reglas de identificación de conceptos antes definidas. En este caso se hizo una generación manual de modelos conceptuales a partir de textos, de modo que se identificó los patrones de *POS Tags* que utiliza un analista para determinar relaciones, la herramienta emula esto y toma los *POS Tags* que se encuentran entre dos sustantivos y cumplen con la regla de mapeo. Así, si por ejemplo, en el archivo se configuran los siguientes patrones:

- pos = adjective, pos = adposition, pos = verb.
- pos = verb, pos = adposition.

Se obtienen los siguientes resultados como una relación, estando estos entre dos *nouns*:

- “múltiples para analizar”, “suficiente para permitir”.
- “dispuesto en”, “prestados a”.

NOTA: “pos” corresponde a una propiedad en el resultado de *POS Tagging*. Esto se puede ver más adelante en el detalle de pasos para la generación del modelo.

Las reglas de mapeo definidas para el caso de estudio se presentan en la Tabla 2.

Generar salida gráfica: Identificados los conceptos a mostrar y sus relaciones, se pasan en forma automática a un formato gráfico, de modo que pueda ser visualizado en una herramienta de software. En

Tabla 2. Reglas de mapeo definidas para el caso de estudio.

Elemento en Modelo	Elemento en Texto	Nota
Concepto	noun	Basadas en la propiedad POS en el XML
Relación	pos = adjective, pos = adposition, pos = verb	Patrones que se identifican en el XML. Se definen propiedad y valor del XML. Patrón “verboconjugado” genera una acción codificada en la herramienta.
	pos = verb, pos = adposition	
	pos = adposition, pos = verb	
	pos = adposition	
	form = que	
	verboconjugado	

el caso de estudio en particular se obtiene como output, para cada documento, un modelo gráfico.

Fase 5. Evaluación

En esta fase se evalúan los resultados del trabajo siguiendo los parámetros definidos en la sección VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN. El caso particular presenta como output el Análisis de Resultados, que corresponde a la sección Análisis de resultados.

IMPLEMENTACIÓN DE HERRAMIENTA PARA AUTOMATIZAR EL MÉTODO

A continuación, se resumen las actividades que permitieron automatizar el método por medio de la implementación de una herramienta. El usuario final ejecuta las actividades en la herramienta en forma secuencial y automática, cada vez que se procesa un documento para generar un modelo conceptual a partir de la información identificada.

Extraer texto de PDF

El texto se extrae mediante el uso de la librería PDFBox 2.0.11 para Java. En este sentido se exploraron varias herramientas gratuitas, (PDFBox para Java y PDFMiner.six para Python), sin embargo, la primera entrega texto más limpio, en la medida que se obtuvo con menos caracteres especiales y menos saltos de página. Lo anterior también motivó el uso de Java como lenguaje de programación para la implementación final.

Pre-procesar texto

En esta actividad se toma el texto obtenido del PDF y se le retira texto innecesario, en particular encabezados y pies de página, líneas de texto de Tablas y líneas de texto de Figuras.

Generar POS Tags

En esta actividad se envía el texto a la herramienta *Freeling 4.0*, la cual retorna un XML con cada

palabra categorizada en una parte de oración (como sustantivo, verbo, adjetivo, etc.). El *POS Tagger* fue seleccionado tomando como referencia los ya usados por los trabajos presentados en el estado del arte que usaban *POS Tagging*, eligiendo entre *Stanford Log-linear Part-Of-Speech Tagger*, Paquete NLTK para Python, *Freeling* y *TreeTagger CIS*.

Se hizo una comparación tomando como referencia dos textos para verificar la precisión de cada una de las herramientas *POS Tagger*. Se ejecutó un proceso de *POS Tagging* manual con el apoyo del diccionario de la Real Academia Española y se comparó con cada herramienta, obteniendo los resultados mostrados en la Tabla 3, que evidencian que el tagger que mejor rendimiento presentó es *Freeling*, por lo tanto, este es el escogido para este trabajo.

Procesar POS Tags

En esta actividad se toma el XML y se recorre para identificar en primera instancia los conceptos y luego las relaciones entre los mismos. Para esto deben estar definidas las reglas de identificación de conceptos y de relaciones en archivo plano (como se mencionó en la fase 3 del método propuesto). Así, al procesar el XML se identifican las secuencias de palabras que cumplen con estos patrones y se toman como elementos válidos.

Además de lo anterior, se obtiene el *Term Frequency (TF)* de cada concepto contando el número de ocurrencias en el XML. Se toma su *Information content (IC)* del archivo plano insumo y se calcula el TFIC para cada concepto, así $TFIC = TF * IC$.

Obtenido este, se ordenan los conceptos por TFIC de mayor a menor, y se identifican como “mostrables en el modelo gráfico” los primeros N conceptos. El valor de N se calcula inicialmente con un número arbitrario asignado debido a experimentación multiplicado por el número de páginas, y el usuario puede cambiarlo si así lo desea.

Tabla 3. Resultado de comparación de *POS Taggers*.

		Stanford	NLTK	Freeling	TreeTagger
Cantidad de palabras correctas	Texto 1	120	46	120	120
	Texto 2	149	61	161	151
Porcentaje con respecto al total de palabras	Texto 1	85%	32%	85%	85%
	Texto 2	85%	35%	91%	86%

- Generar salida gráfica

Identificados los conceptos y relaciones, se convierten éstos en salidas gráficas. Se hizo una exploración de herramientas gratuitas que permiten generar modelos conceptuales y que además permitan recibir un formato escrito en texto, en lugar de o además de generación gráfica directamente por un usuario.

Con lo anterior se identificó la herramienta yEd Graph Editor, que permite mostrar gráficamente modelos escritos en formato *graphml*. Esta herramienta permite hacer zoom al diagrama generado, lo que resulta útil para modelos de textos extensos, además habilita al usuario para editarlo gráficamente, modificando, eliminando o adicionando tanto conceptos (clases para la herramienta) como relaciones.

En la Figura 6 se muestra la visualización para el texto de ejemplo. Los conceptos en fondo azul y letra blanca corresponden a los cinco conceptos con mejor TFIC de acuerdo al ordenamiento explicado anteriormente. Esto permite al usuario identificar los conceptos con mejor peso para el modelo y a partir de ahí hacer el análisis del modelo generado. Esto es útil en modelos resultantes con gran cantidad de conceptos.

Además de la herramienta antes mostrada, se puede usar la herramienta *Mermaid Live Editor*, este es un proyecto que se encuentra publicado en la *GitHub* y permite, a partir de un código específico, generar

modelos. Esta, a diferencia de yEd Graph Editor no siempre permite hacer un zoom consistente que permita leer claramente el modelo resultante, además, no permite adicionar atributos a los conceptos, así, en un hipotético trabajo futuro en el que se adiciona atributos a los conceptos no se podría usar esta herramienta.

VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

Diseño experimental

En esta sección se presentan los elementos principales del experimento a ejecutar para la validación de la propuesta.

Definición de variables

La variable independiente definida es la aplicación SI/NO de la técnica en la lectura de los textos.

Las variables dependientes, son:

- Comparativo del tiempo que toma procesar, comparando Tiempo aplicando lectura directa con Tiempo aplicando la técnica de NLP. Se mide en minutos desde que inicia el proceso hasta que se finaliza para hacer la comparación.
- Comparativo del nivel de interpretación de la información contenida en los documentos, comparando nivel de interpretación aplicando lectura directa con nivel de interpretación

Figura 6. Ejemplo de visualización de graphml en yEdGraphics.

aplicando la técnica de NLP. Se mide determinando el porcentaje de respuestas correctas en cada documento para la comparación.

Tratamiento de grupos

Se definen dos grupos conformados por ingenieros de requisitos quienes en su labor construyen documentos de requisitos e ingenieros de desarrollo que en su labor usan como insumo documentos de requisitos ya construidos, así:

- Grupo A, denominado grupo base o grupo de control: Ingenieros que toman una serie de documentos escritos en lenguaje natural, leen su contenido y al final contestan un cuestionario para cada documento.
- Grupo B, denominado grupo con enfoque propuesto: Ingenieros que toman los mismos documentos escritos en lenguaje natural, les aplican una o varias herramientas para generar un modelo de representación, al final contestan un cuestionario para cada documento. Para observaciones adicionales se dio un nivel de dificultad a cada pregunta, siendo 1 el nivel más bajo y 3 el nivel más alto.

Calificación de calidad del modelo generado

Si bien la ejecución del cuestionario permite verificar el nivel de comprensión, resulta útil medir otros aspectos que determinen la calidad de los modelos resultantes, es por esto que se tomó el estándar ISO/IEC 9126-3 en el modelo de datos conceptual entidad-relación [41]. Luego de contestar el cuestionario, todos los analistas obtienen el modelo de cada documento y lo califican de 1 a 5 siendo 1 la peor calificación, 5 la mejor calificación, aplicando esta medida a

cada uno de los siguientes criterios: Legibilidad, Completitud, Corrección, Minimalidad, Expresividad, Autoexplicación. Estos están definidos por la norma ISO/IEC 9126-3 en la Tabla 4.

Criterios de satisfacción

La hipótesis se comprueba si se cumple las siguientes premisas con respecto a las variables dependientes: El tiempo que toma procesar los documentos al grupo B es inferior al tiempo que toma procesar los documentos aplicando lectura directa.

1. El porcentaje de respuestas correctas luego de aplicar la técnica de NLP es igual o superior al porcentaje de respuestas correctas aplicando lectura directa.
2. Se deben cumplir ambas premisas, si se cumple solo una la hipótesis no puede ser comprobada.

ESPECIFICACIONES DE LA VALIDACIÓN

En la validación se consideran los siguientes 4 documentos de resoluciones:

1. Resolución 5111 de 2017. Se toman las primeras 8 páginas.
2. Proyecto de resolución 5397 de 2018.
3. Proyecto de resolución 4930 de 2016.
4. Proyecto de resolución 5161 de 2016.

Estos son documentos de regulación de comunicaciones expedidos por la Comisión de Regulación de Comunicaciones de Colombia (CRC). Dado que son de dominio público expuestos directamente en su portal web <https://www.crcm.gov.co/> o en el portal web del Ministerio de Comunicaciones <https://www.mintic.gov.co>, están escritos en lenguaje natural

Tabla 4. Criterios de calidad para modelos conceptuales. Tomado de: ISO/IEC 9126-3 [41].

Criterio	Descripción
Legibilidad	Está enfocado a las consideraciones visuales para la lectura y presentación del modelo conceptual.
Completitud	El modelo debe incluir lo que se quiere diseñar, qué es aquello que se encuentra plasmado en los requerimientos del sistema por desarrollar.
Corrección	Cada elemento se representa haciendo uso de las estructuras adecuadas.
Minimalidad	Un modelo conceptual se considera mínimo si no tiene información redundante o duplicada.
Expresividad	El modelo representa la realidad, de manera que con sus elementos esta puede ser comprendida fácilmente.
Autoexplicación	La lógica del negocio con respecto a los datos puede ser accedida y entendida por el modelo conceptual.
Extensibilidad	Capacidad de un esquema para poder tolerar cambios y adaptarse a nuevas necesidades de los usuarios.

en idioma español, y son usados por ingenieros de requisitos para su contextualización, de modo que los productos de software para las empresas que prestan servicios de telecomunicaciones cumplan con la regulación existente.

HALLAZGOS

Resultados

En la Tabla 5 se muestran los resultados del tiempo que tomó el proceso manual desde que el analista inició hasta que contestó las preguntas formuladas. Se incluyen en las últimas columnas los porcentajes de respuestas correctas.

Prueba de hipótesis

Para comprobar la hipótesis se ejecuta una prueba de hipótesis así:

- En este caso se están comparando dos poblaciones que contienen 16 datos cada una, al ser un número inferior a 30 se utiliza la distribución t-Student para la prueba.
- Las variables para la definición de hipótesis son las siguientes:

TU1 = tiempo muestra grupo A o grupo de control

TU2 = tiempo muestra grupo B o grupo con el método aplicado

PU1 = porcentaje correctas muestra grupo A o grupo de control

Tabla 5. Resultados de las evaluaciones de los documentos.

Documento	Analista	Grupo	Tiempo Minutos	% Correctas Nivel 1	% Correctas Nivel 2	% Correctas Nivel 3	% Correctas Total
1	JJ	A	57	83,33%	100,00%	75,00%	86,67%
1	DR	A	48	83,33%	100,00%	100,00%	93,33%
1	JM	A	73	83,33%	100,00%	100,00%	93,33%
1	JS	A	64	100,00%	100,00%	50,00%	86,67%
2	JJ	A	35	83,33%	100,00%	100,00%	90,00%
2	DR	A	37	100,00%	100,00%	100,00%	100,00%
2	JM	A	51	100,00%	100,00%	100,00%	100,00%
2	JS	A	25	83,33%	100,00%	100,00%	90,00%
3	JJ	A	54	100,00%	100,00%	100,00%	100,00%
3	DR	A	47	100,00%	100,00%	100,00%	100,00%
3	JM	A	62	75,00%	85,71%	100,00%	85,71%
3	JS	A	49	75,00%	85,71%	100,00%	85,71%
4	JJ	A	25	100,00%	33,33%	66,67%	62,50%
4	DR	A	28	100,00%	66,67%	66,67%	75,00%
4	JM	A	47	100,00%	66,67%	100,00%	87,50%
4	JS	A	21	100,00%	66,67%	66,67%	75,00%
1	LZ	B	23	100,00%	100,00%	100,00%	100,00%
1	ES	B	45	83,33%	100,00%	100,00%	93,33%
1	OO	B	40	100,00%	80,00%	75,00%	86,67%
1	JR	B	52	83,33%	80,00%	75,00%	80,00%
2	LZ	B	18	83,33%	100,00%	0,00%	80,00%
2	ES	B	32	100,00%	66,67%	100,00%	90,00%
2	OO	B	25	100,00%	100,00%	100,00%	100,00%
2	JR	B	18	100,00%	100,00%	100,00%	100,00%
3	LZ	B	19	100,00%	71,43%	100,00%	85,71%
3	ES	B	28	100,00%	85,71%	66,67%	85,71%
3	OO	B	30	100,00%	85,71%	66,67%	85,71%
3	JR	B	21	100,00%	85,71%	100,00%	92,86%
4	LZ	B	13	100,00%	66,67%	100,00%	87,50%
4	ES	B	16	50,00%	100,00%	33,33%	62,50%
4	OO	B	15	100,00%	100,00%	100,00%	100,00%
4	JR	B	17	50,00%	100,00%	33,33%	62,50%

PU2 = porcentaje correctas muestra grupo B o grupo con el método aplicado

- De acuerdo a la definición dada en el ítem “Criterios de satisfacción”, la hipótesis nula (H_0) y la hipótesis alternativa (H_1) se definen así:

H_0 : ($TU1 \leq TU2$) o ($PU1 > PU2$)

H_1 : ($TU1 > TU2$) y ($PU1 \leq PU2$)

- El nivel de significancia o alfa se define como 0.05. La literatura recomienda un valor entre 0.01 y un 0.05. Considerando la naturaleza del experimento y el rigor con el que se abordan la captura de tiempos y el diligenciamiento de los cuestionarios, se opta por el valor 0.05.
- La validez estadística de las conclusiones de las pruebas de hipótesis se garantiza mediante un nivel de confianza del 95% y se buscaron muestras representativas de acuerdo con las posibilidades, intentando evitar aceptar una hipótesis que fuera falsa o rechazar una que fuera verdadera.
- Estas pruebas se realizan mediante el complemento de Análisis de Datos de Microsoft Excel 2016. Para esta se asume que ambas muestras tienen varianzas desiguales, por lo que se ejecuta la función “Prueba t para dos muestras suponiendo varianzas desiguales”.
- La herramienta solicita la diferencia hipotética entre las medidas, para el caso del ejemplo asumimos que las dos muestras no tienen diferencia hipotética, dado que es precisamente lo que se busca establecer con la prueba, así, el valor para este parámetro es cero (0).
- Dado que la hipótesis usa dos variables de decisión y la herramienta sólo permite comparar dos muestras cada vez, se ejecutan dos pruebas t: la primera se usa para comparar las muestras teniendo en cuenta primero el tiempo ($TU1$ y

$TU2$) y la segunda para comparar el porcentaje de respuestas correctas ($PU1$ y $PU2$).

- Los resultados muestran varios datos, entre los cuales se tienen el *valor crítico de t* (una cola), *Valor crítico de t* (dos colas) y el *Estadístico t*. De acuerdo a los valores y la definición, se evalúan resultados con respecto a la hipótesis así:
 - En la primera prueba t el *valor crítico t de una cola* se ubica a la derecha, es decir positivo. Para rechazar la hipótesis nula con respecto al tiempo, el *estadístico t* debe ser mayor al *valor crítico t* positivo. En la Figura 7 se ejemplifica la evaluación.

En la segunda *prueba t* el *punto crítico t de una cola* se ubica a la izquierda, es decir negativo. Para rechazar la hipótesis nula con respecto al porcentaje de respuestas correctas, el *estadístico t* debe ser menor al *punto crítico t* negativo. En la Figura 8 se ejemplifica la evaluación.

Resultado de la prueba

Ejecutada la prueba t con respecto al tiempo se obtuvo valor *t crítico* = 1,7011309, y *Estadístico t* = 4,0502712, con estos datos es posible afirmar que para el tiempo:

H_0 : ($TU1 \leq TU2$). Se rechaza.

H_1 : ($TU1 > TU2$). Se comprueba.

Esto solo evidencia la primera parte de la hipótesis, así para el porcentaje de respuestas correctas sin diferenciar por nivel de dificultad, es decir tomando todas, se obtuvo un valor *t crítico* = 1,7011309, y *Estadístico t* = 4,0502712. Con estos datos es posible afirmar que para el porcentaje de respuestas correctas:

H_0 : ($PU1 > PU2$). No se rechaza.

H_1 : ($PU1 \leq PU2$). Se rechaza.

Figura 7. Región de aceptación o rechazo de H_0 , cola derecha.

Figura 8. Región de aceptación o rechazo de H_0 , cola izquierda.

De acuerdo a lo anterior y complementando las hipótesis inicialmente planteadas, se define que:

H_0 : ($TU1 \leq TU2$) o ($PU1 > PU2$). No se rechaza.

H_1 : ($TU1 > TU2$) y ($PU1 \leq PU2$). Se rechaza.

En la validación de porcentajes de respuestas correctas por nivel se encontró un comportamiento igual, como se describe en la Tabla 6.

Calificación de los modelos basada en criterios

Además de los datos propios para comprobar la hipótesis, los analistas dieron calificaciones de cada documento, de acuerdo a los criterios definidos en el capítulo anterior, generando los resultados presentados en la Tabla 7.

ANÁLISIS DE RESULTADOS

La primera variable a tener en cuenta para la hipótesis es el tiempo, en este sentido se evidenció claramente que el tiempo para procesamiento usado por el grupo de control es superior al tiempo tomado por el grupo con la aplicación del método, tal como se comprueba en las pruebas de hipótesis.

Es importante anotar que los analistas interpretan previamente el modelo, antes de proceder a responder las preguntas, proceso que resultaba rápido, sin embargo, al contestar las preguntas tenían que apoyarse en el modelo sin una opción de búsqueda de conceptos por lo que tenían que hacer una búsqueda

Tabla 6. Resultados para porcentajes de correctas de niveles 1, 2 y 3.

Nivel pregunta	Valor t crítico	Estadístico t	Estado
1	-1,71088208	0,209121444	H_0 no se rechaza
2	-1,71088208	-0,1825003	H_0 no se rechaza
3	-1,71088208	1,22128806	H_0 no se rechaza

Tabla 7. Resultados de calificaciones de los modelos resultantes.

	Promedio grupo A ↓	Promedio Grupo B ↓	Promedio total ↓
Legibilidad	3,31	3,56	3,44
Compleitud	3,94	3,56	3,75
Corrección	3,69	3,94	3,81
Minimalidad	3,56	3,94	3,75
Expresividad	4,13	3,63	3,88
Autoexplicación	3,75	3,63	3,69
Extensibilidad	3,88	4,06	3,97
Promedio →	3,75	3,76	3,75

manual, que en muchos casos resultaba intuitiva. Con la lectura de los documentos las respuestas se podían contestar mucho más rápidamente al poder hacer búsqueda de términos.

Con respecto al porcentaje de respuestas correctas se dividió las preguntas en tres niveles de dificultad de modo que se pudiera evaluar el método propuesto de acuerdo a estos niveles, sin embargo, para todos los casos se observó un comportamiento similar en tanto la prueba de hipótesis *tipo t* para cada caso no permitió concluir que se acepta la hipótesis planteada en este documento.

Si se tienen en cuenta los promedios en los porcentajes de respuestas correctas, se puede establecer un comparativo entre los resultados del grupo A y el grupo B, encontrando que las diferencias porcentuales para las preguntas de nivel de dificultad uno y dos son cercanas a 1%, lo que muestra que el método propuesto permite mantener un nivel de interpretación muy cercano al proceso realizado en forma directa mediante lectura de documentos para cuestiones básicas. Esto no sucede para preguntas de nivel 3 donde la diferencia es más significativa, como se ve en la Tabla 8. Con esto, y teniendo en cuenta que el tiempo efectivamente se reduce aplicando el método, este ofrece una buena aproximación a la hipótesis planteada.

En la calificación dada por los analistas, se observa que en general no se presentan grandes diferencias entre las evaluaciones que se emitieron ambos grupos, anotando que la legibilidad para ambos es el aspecto con la calificación más baja, lo que muestra que este aspecto es el primero que se debe considerar para una evolución de la herramienta implementada.

En general, las calificaciones son consistentes con los resultados para los porcentajes de respuestas correctas, en tanto no presenta los resultados más altos, sin embargo, se obtienen resultados en niveles que permiten concluir que el método es una buena

aproximación a solucionar el problema planteado en este trabajo, que requiere de mejoras para establecerse como una solución definitiva.

CONCLUSIONES Y TRABAJO FUTURO

Conclusiones

- La industria presenta retos para el procesamiento de información de negocio contenida en documentos textuales. Esta información puede ser útil para los ingenieros de requisitos, sin embargo, no siempre la usan para su labor por lo que resulta útil proveer herramientas que faciliten esta tarea.
- Es posible aplicar técnicas de procesamiento de lenguaje natural que permitan procesar texto y deriven en la generación de modelos de representación del conocimiento, así, la comunidad científica ha propuesto métodos y/o herramientas de software cuyo objetivo es extraer información de texto y generar representación de conocimiento.
- Si bien se evidencian propuestas, existen pocos trabajos que busquen procesar información en idioma español, y que además generen algún modelo de representación de conocimiento, por lo que, en el ámbito regional como países de habla hispana, resulta muy útil presentar un método que atienda este segmento.
- De acuerdo a la experimentación, es útil representar conocimiento en modelos ya conocidos por las comunidades en las que influye el dominio procesado o que sean fácilmente interpretables, como lo son los modelos UML.
- Es de gran ayuda aplicar técnicas o métodos ya propuestos que permitan organizar las actividades a ejecutar, como por ejemplo en el caso particular, el método para proyectos de analítica de texto permitió organizar las actividades para aplicar NLP a documentos de negocio generando representación del conocimiento.
- El método propuesto, si bien no presenta una solución definitiva al problema planteado, de

Tabla 8. Resultados de calificaciones de los modelos resultantes.

Grupo	% Correctas Nivel 1	% Correctas Nivel 2	% Correctas Nivel 3	% Correctas Total
A	91,67%	87,80%	89,06%	88,21%
B	90,63%	88,87%	78,13%	87,03%
Diferencia(A-B)	1,04%	-1,07%	10,94%	1,18%

acuerdo a los resultados se demuestra que es útil y aplicable para el caso de estudio presentado, y podría ser implementado y probado en otros dominios.

- Si bien la evaluación del método arrojó resultados aceptables, es necesario evolucionar de modo que permita mejorar la calidad del conocimiento representado con respecto al contenido de los documentos procesados, tanto para el dominio presentado en el caso de estudio como para otros dominios en los cuales pueda ser aplicado.

TRABAJO FUTURO

- Afinar el método para que priorice las relaciones identificadas y tome sólo la más relevante, evitando poner más de una relación entre dos atributos.
- Adicionar al método la identificación de atributos de los conceptos identificados a partir del documento, así como su implementación en la herramienta de software, para que se identifique y se grafique en el modelo resultante.
- Verificar resultados en términos del apoyo a la tarea de generación de requisitos basados en modelos conceptuales resultantes de la aplicación del método acá propuesto.
- Evaluar la calidad de los modelos generados por la herramienta al aplicar el método.
- Pre-entrenar *Freeling* para afinar la identificación de conceptos de negocio específicos o que *Freeling* no identifique correctamente.
- Incluir a los analistas que participan de la generación automática de modelos en la identificación de *IC (Information Content)* para los pesos de negocio que sirven de entrada a la herramienta de software.
- Dado que muchos de los documentos son de gran extensión, sería útil hacer una “descomposición funcional”, es decir, que el modelador identifique grupos de conceptos en forma automática y genere varios modelos de una vez, de acuerdo a esta agrupación.
- Mejorar gráficamente el modelador de modo que se presente un *Wizard*, lo que brindaría una mejor experiencia de usuario y permitiría acciones adicionales, por ejemplo, seleccionar específicamente los conceptos que desea ver en el modelo.

- Integrar una de las herramientas de visualización al modelador mismo, así se evitaría tener que instalar y/o usar otra herramienta de software.

REFERENCIAS

- [1] P. Velasco-Elizondo, R. Marín-Piña, S. Vazquez-Reyes, A. Mora-Soto and J. Mejia. “Knowledge representation and information extraction for analyzing architectural patterns”. *Sci. Comput. Program.* Vol. 121, pp. 176-189. 2016.
- [2] C. Burnay, I.J. Jureta and S. Faulkner. “What stakeholders will or will not say: A theoretical and empirical study of topic importance in Requirements Engineering elicitation interviews”. *Inf. Syst.* Vol. 46, pp. 61-81. 2014.
- [3] N. Ibrahim, W. Kadir, M.N. Wan and S. Deris. “Documenting requirements specifications using natural language requirements boilerplates”, pp. 19-24. 2014.
- [4] V. Arnaoudova, S. Haiduc, A. Marcus and G. Antoniol. “The Use of Text Retrieval and Natural Language Processing in Software Engineering”. In 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, pp. 949-950. 2015.
- [5] W. Hsu, C.W. Arnold and R.K. Taira. “A Neuro-Oncology Workstation for Structuring, Modeling, and Visualizing Patient Records”. In *IHI'10 Proceedings of the 1st ACM International Health*, pp. 837-840. 2010.
- [6] K.M. Annervaz, V. Kaulgud, S. Sengupta and M. Savagaonkar. “Natural language requirements quality analysis based on business domain models”. In 2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 676-681. 2013.
- [7] G. Ninaus, A. Felfernig, M. Stettinger and S. Reiterer. “INTELLIREQ: Intelligent Techniques for Software Requirements Engineering”, pp. 1161-1166. 2014.
- [8] S.M. Jimenez, A. Gelbukh and G. Sidorov. “Generating summaries by means of synthesis of conceptual graphs”. *Rev. Signos.* Vol. 47, Issue 86, pp. 463-485. 2014.
- [9] J.F. Allen. “Natural language processing”, pp. 1218-1222. 2003.

- [10] G.G. Chowdhury. "Natural Language Processing". Vol. 37, Issue 1, pp. 51-89. 2003.
- [11] M. Ozkaya. "What is software architecture to practitioners: A survey". 2016 4th Int. Conf. Model. Eng. Softw. Dev., pp. 677-686. 2016.
- [12] W. Daelemans. "POS Tagging". In Encyclopedia of Machine Learning and Data Mining, T. Zeugmann, Ed. 2017.
- [13] D.W. Embley and B. Thalheim. "Handbook of Conceptual Modeling". 2011.
- [14] C. Larman. "UML y patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado". 2nd ed. Madrid: PEARSON. 2003.
- [15] M. Brass and Y. Toussaint. "Artificial intelligence tools for software engineering: Processing natural language requirements". Trans. Inf. Commun. Technol. Vol 2, pp. 275-290. 1995.
- [16] B. Javed and S. Sultan M. "Process Support for Requirements Engineering Activities in Global Software Development: A Literature Based Evaluation". 2010 Int. Conf. Comput. Intell. Softw. Eng., pp. 1-6. 2010.
- [17] X. Xiao, A. Paradkar and T. Xie. "Automated extraction and validation of security policies from natural-language documents". Proc. ACM SIGSOFT 20th Int. Symp. Found. Softw. Eng. 2012.
- [18] O.A. Beltrán G. "Revisiones sistemáticas de la literatura". Vol. 20 N° 1, pp. 60-69. 2005.
- [19] T. Diamantopoulos, M. Roth, A. Symeonidis and E. Klein. "Software Requirements as an Application Domain for Natural Language Processing". Lang. Resour. Eval. Vol. 51, pp. 495-524. 2017.
- [20] U. Iqbal and I.S. Bajwa. "Generating UML activity diagram from SBVR rules". In 2016 Sixth International Conference on Innovative Computing Technology (INTECH), pp. 216-219. 2016.
- [21] V. Bhala, V. Sagara and S. Abiramib. "Conceptual modeling of natural language functional requirements". J. Syst. Softw. Vol. 88, Issue 1, pp. 25-41. 2014.
- [22] D.A. de Araujo, S.J. Rigo, C. Muller and R. Chishman. "Automatic information extraction from texts with inference and linguistic knowledge acquisition rules". 2013 IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol. Vol. 3, pp. 151-154. 2013.
- [23] S. Roychoudhury, N. Bellarykar and V. Kulkarni. "A NLP Based Framework to Support Document Verification-as-a-Service". In 2016 IEEE 20th International Enterprise Distributed Object Computing Conference (EDOC), pp. 139-148. 2016.
- [24] I.S. Bajwa, B. Bordbar and M. Lee. "OCL usability: a major challenge in adopting UML". In Proceedings of the 3rd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering - RAISE 2014. Vol. 92, pp. 32-37. 2014.
- [25] R. Pinquie, P. Véron, F. Segonds and N. Croué. "Natural Language Processing of Requirements for Model-Based Product Design with ENOVIA / CATIA V6". PLM 2015 Prod. Lifecycle Manag. Era Internet Things, pp. 205-215. 2015.
- [26] K.P. Sawant, S. Roy, D. Parachuri, F. Plesse and P. Bhattacharya. "Enforcing structure on textual use cases via annotation models". In ISEC '14 Proceedings of the 7th India Software Engineering Conference. 2014.
- [27] P. Fernandes, L.O.C. Furquim and L. Lopes. "A supervised method to enhance vocabulary with the creation of domain specific lexica". In 2013 IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT). Vol. 3, pp. 139-142. 2013.
- [28] C.P. Escartín and H.M. Alonso. "Choosing a Spanish Part-of-Speech tagger for a lexically sensitive task". Proces. del Leng. Nat. Rev. Vol. 54, Issue 54, pp. 29-36. 2015.
- [29] B. Hnatkowska and T. Gawęda. "Automatic Processing of Dynamic Business Rules Written in a Controlled Natural Language". Towar. a Synerg. Comb. Res. Pract. Softw. Eng. Vol. 3, pp. 91-103. 2018.
- [30] H. Van der Aa, H. Leopold and H.A. Reijers. "Comparing textual descriptions to process models - The automatic detection of inconsistencies". Inf. Syst. Vol. 64, pp. 447-460. 2017.
- [31] S. Miranda, F. Orciuoli and D.G. Sampson. "A SKOS-based framework for Subject Ontologies to improve learning experiences". Comput. Hum. Behav. Vol. 61, pp. 609-621. 2016.

- [32] D. Movshovitz-Attias and W.W. Cohen. "Natural Language Models for Predicting Programming Comments". In *ACL*, pp. 35-40. 2013.
- [33] M.A. Naeem and I.S. Bajwa. "Generating OLAP queries from natural language specification". In *ICACCI '12 Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, pp. 768-773. 2012.
- [34] B. Keith Norambuena, E. Fuentes Lettura and C. Meneses Villegas. "Sentiment analysis and opinion mining applied to scientific paper reviews". *Intell. Data Anal.* Vol. 23, pp. 191-214. 2019.
- [35] C. Miranda Henríquez y J. Guzmán, "Extracción de información desde la web para identificar acciones de un modelo de dominio en planificación automática". *Ingeniare. Revista chilena de ingeniería*. Vol. 23 N° 3, pp. 439-448. 2015.
- [36] J.L. Jara, M. Chacón and G. Zelaya. "Empirical evaluation of three machine learning methods for automatic classification of neoplastic diagnoses". *Ingeniare. Revista chilena de ingeniería*. Vol. 19 N° 3, pp. 359-368. 2011.
- [37] B. Manrique, J.B. Quintero, D.A. Marin y J.S. Morales. "Analítica de Texto para procesar documentos de requisitos". En *Tecnologías del lenguaje humano: aplicaciones desde la Lingüística Computacional y de Corpus*, Sello Editorial Universidad de Medellín, Ed. Medellín. 2018.
- [38] J. Wiley. "Data Science and Big Data Analytics". 2015.
- [39] A. Olivé. "Conceptual Modeling of Information Systems". 2007.
- [40] D. Rosenberg and M. Stephens. "Domain Modeling". In *Use Case Driven Object Modeling with UML*, pp. 23-48. 2007.
- [41] M.F. González Pinzón y J.S. González Sanabria. "Aplicación del estándar ISO/IEC 9126-3 en el modelo de datos conceptual entidad-relación". *Revista Facultad de Ingeniería, UPTC*. Vol. 22 N° 35. pp. 113-125. 2013.