

# Una Comparación de Técnicas de NLP Semánticas para Analizar Casos de Uso

Alejandro Rago<sup>\*†1</sup>, Claudia Marcos<sup>\*‡2</sup>, J. Andrés Díaz-Pace<sup>\*†3</sup>

<sup>\*</sup>*Instituto Superior de Ingeniería de Software (ISISTAN-UNICEN)*

*Tandil, Buenos Aires, Argentina*

<sup>†</sup>*CONICET, Argentina*

<sup>‡</sup>*CIC, Buenos Aires, Argentina*

<sup>1 2 3</sup>{arago, cmarcos, adiaz}@exa.unicen.edu.ar

**Resumen**—El análisis de documentos textuales escritos en lenguaje natural ya no se considera una tarea compleja, gracias a los avances en técnicas de Procesamiento de Lenguaje Natural (NLP). Sin embargo, ciertas aplicaciones requieren un análisis semántico más profundo para obtener buenos resultados. En este artículo, se presenta un estudio exploratorio de técnicas de NLP semánticas para descubrir concerns latentes en especificaciones de casos de uso. Con este fin, se proponen dos técnicas de NLP: clustering semántico y reglas enriquecidas semánticamente. Los resultados de la evaluación experimental de estas dos técnicas, y su comparación contra una tercera técnica desarrollada por otros investigadores, muestran que las técnicas de NLP semánticas tienen un gran potencial para detectar concerns candidatos. Particularmente, si estas técnicas son configuradas correctamente, pueden ayudar a reducir el esfuerzo de los analistas de requerimientos y promover la construcción de software de mejor calidad.

**Abstract**—The analysis of textual documents written in natural language has become feasible thanks to the advances in Natural Language Processing (NLP) techniques. However, certain applications require a deeper semantic analysis of the text to produce good results. In this article, we present an exploratory study of semantic-aware NLP techniques for discovering latent concerns in use case specifications. To this end, we propose two NLP techniques, namely: semantic clustering and semantically-enriched rules. After evaluating these two techniques and comparing them with a technique developed by other researchers, results have showed that semantic NLP techniques hold great potential for detecting candidate concerns. Particularly, if these techniques are properly configured, they can help to reduce the efforts of requirement analysts and promote better quality in software development.

## I. INTRODUCCIÓN

En la actualidad, existe un interés creciente en la aplicación de tecnologías para el Procesamiento de Lenguaje Natural (Natural Language Processing, NLP) a la industria. El NLP permite procesar información descrita en lenguaje natural, analizando tanto sus propiedades léxicas y sintácticas (por ej., cuestiones gramaticales) como así también semánticas (por ej., el significado de una frase). Las técnicas de NLP utilizan métodos de inteligencia artificial (Artificial Intelligence, AI), de Recuperación de Información (Information Retrieval, IR) [1] y de aprendizaje supervisado (Machine Learning, ML) [2] para realizar los análisis lingüísticos. En los últimos años, se han efectuado importantes avances en el desarrollo de técnicas de NLP semánticas capaces de aproximar el significado e intención de las oraciones [3], [4].

En el contexto de la Ingeniería de Software, los documentos textuales son usados frecuentemente como medio de comunicaciones entre los stakeholders [5] y sirven para comunicar requerimientos, decisiones arquitectónicas y casos de prueba. Por esta razón, el NLP es una alternativa interesante para automatizar ciertas actividades del ciclo de vida del software. Especialmente, ciertos problemas que ocurren en etapas tempranas del desarrollo y son críticos para el éxito de un proyecto pueden ser mitigados con herramientas equipadas con tecnologías NLP [6]. Uno de estos problemas está relacionado con los requerimientos de software. Las especificaciones de casos de uso, por ejemplo, prestan demasiada atención a la funcionalidad del sistema y pueden causar que los analistas no tengan en cuenta otros *concerns* importantes que son de naturaleza “no funcional” [7]. Asimismo, los casos de uso proveen un soporte limitado para capturar dichos concerns apropiadamente. En la literatura, este tipo de concerns son denominados Cross-cutting Concerns (CCCs) [8]. En la práctica, es normal que los analistas desestimen las guías y recomendaciones para especificar casos de usos e incorporen información acerca de los CCCs de forma desordenada en las especificaciones textuales. A pesar de sus buenas intenciones (evitar dejar concerns sin documentar), no es una solución ideal y tiene un número de consecuencias negativas. Principalmente, los casos de uso se tornan difíciles de leer, debido a que casos de uso individuales incluyen concerns entremezclados y los CCCs terminan dispersos en múltiples documentos. Aunque se han desarrollado enfoques para “minar” estos concerns [9]–[11] combinando técnicas de NLP y IR, el pobre análisis semántico de estos enfoques dificulta el descubrimiento de gran parte de los requerimientos afectados por los CCCs.

En este artículo, se estudia la aplicación de técnicas de NLP semánticas para contribuir a la detección de concerns latentes en las especificaciones de casos de uso. Nuestra hipótesis es que las técnicas de NLP semánticas son un mecanismo eficaz para identificar CCC desde el comienzo del desarrollo, mejorando así la comprensión de los requerimientos. Con este fin, se desarrollaron dos técnicas semánticas diferentes. La primera técnica combina NLP, algoritmos de clustering y métricas de similitud entre palabras para encontrar comportamientos repetidos que pueden ser potenciales CCCs. La segunda técnica enriquece mecanismos de búsqueda basados en reglas con información semántica de las oraciones para revelar “familias” de concerns mediante consultas específicas. Las contribuciones de este trabajo

son: (i) la definición de técnicas que se benefician de la semántica para encontrar CCCs, y (ii) el estudio empírico de dichas técnicas de NLP en un dominio particular de la Ingeniería. Las técnicas fueron evaluadas en tres casos de estudio comparando sus resultados con aquellos obtenidos por un grupo de analistas expertos. A modo comparativo, también se analizaron los resultados de una técnica para buscar CCC desarrollada por terceros que no aprovecha la semántica del texto. Los resultados de los experimentos fueron prometedores, y las técnicas semánticas mostraron una buena *precision* y *recall*, especialmente con las reglas. Asimismo, las técnicas propuestas mejoraron los resultados de la técnica de terceros por un margen considerable.

El resto del artículo está organizado de la siguiente forma. La sección II discute trabajos que aplican NLP para extraer información de los requerimientos. La sección III presenta las técnicas de NLP semánticas desarrolladas para detectar concerns. La sección IV está dedicada a la evaluación experimental y la discusión de los resultados. Por último, la sección V da las conclusiones del trabajo, mencionando limitaciones y enumerando líneas de trabajo futuras.

## II. TRABAJOS RELACIONADOS

La utilización de diversas técnicas de NLP, ML, e incluso AI para procesar artefactos relacionados al software es una práctica común en estos días. En la industria, aproximadamente el 80 % de la documentación de requerimientos es descrita usando lenguaje natural [5], por lo que la aplicación de técnicas de NLP es una elección bastante evidente. En el contexto de la Ingeniería de Requerimientos (Requirements Engineering, RE), herramientas capaces de interpretar el texto de los documentos y extraer información son de gran ayuda para simplificar el trabajo de los analistas [6].

En este trabajo, se analizaron aquellos trabajos que permiten minar de forma semiautomática concerns latentes en los documentos y relevantes para el desarrollo. Uno de los primeros trabajos en el área fue de Rosenhainer, el cuál aplicó técnicas de IR para automatizar la búsqueda de CCCs en especificaciones de software [12]. La búsqueda consiste en realizar consultas individuales descritas en forma de expresiones regulares y aplicando técnicas de “string matching”. Posteriormente, Baniassad y Clarke desarrollaron el enfoque Theme como una herramienta de soporte para la orientación a aspectos en los niveles de requerimientos y diseño [9]. Específicamente, Theme/Doc construye vistas de las especificaciones de requerimientos para exponer las relaciones entre los diferentes comportamientos del sistema y así descubrir CCCs. La herramienta, sin embargo, requiere dos entradas manuales: acciones claves y entidades claves, las cuales son usualmente provistas por un analista. Más recientemente, Busyairah y Zarinah desarrollaron el enfoque 3CI en base a las ideas centrales de Theme/Doc [11]. 3CI automatiza aún más la identificación de los CCCs evitando la entrada manual de acciones y entidades claves. Este enfoque depende en varias técnicas de NLP para determinar las dependencias entre los requerimientos, mayormente derivados del análisis de verbos dominantes. La herramienta EAMiner implementa otro enfoque para minar concerns mediante el etiquetado de parte del discurso

(Part-of-Speech, POS) de los requerimientos textuales [10]. Adicionalmente, cada palabra es categorizada utilizando una taxonomía conceptual de la gramática inglesa. Cada categoría es posteriormente asociada a requerimientos no funcionales (NFR). Posteriormente, EAMiner sugiere como CCC cada palabra categorizada y asociada con un NFR.

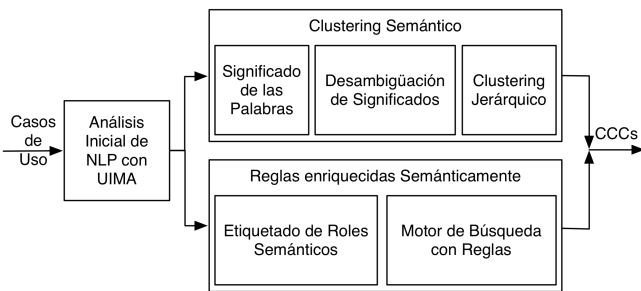
Aunque los trabajos existentes son interesantes, todavía requieren de ciertos ajustes para ser llevados a la práctica. Esto se debe principalmente a la falta de análisis semánticos que permitan “comprender” los requerimientos. El trabajo de Rosenhainer, por ejemplo, tiene deficiencias ante la presencia de sinónimos en el texto. Por otro lado, tanto 3CI como Theme/Doc generan sugerencias incorrectas cuando hay palabras ambiguas en los documentos. Por último, aunque EAMiner intenta paliar algunos de estos problemas con las categorías conceptuales, no deja de ser una solución ad-hoc que debe ser adaptada a cada proyecto en particular para tener buenos resultados.

## III. DESCUBRIENDO CONCERNS EN REQUERIMIENTOS TEXTUALES

Es común que las especificaciones de requerimientos contengan referencias o pistas de los concerns ocultos en el texto. Sabiendo que inspeccionar los documentos es una tarea propensa a errores y que lleva mucho tiempo, aprovechar técnicas basadas en NLP que permitan descubrir CCCs potenciales es una idea razonable. En esta sección, se presentan dos técnicas para descubrir CCCs candidatos que están ocultos en las especificaciones. Estas técnicas comparten dos aspectos claves: ambas incluyen un análisis inicial de los requerimientos utilizando técnicas tradicionales de NLP (es decir, no semánticas) para procesar los requerimientos, y ambas incluyen una serie de técnicas de NLP semánticas para mejorar la detección de concerns. Ambas técnicas tienen en cuenta la intención y significados de los requerimientos para así intentar mitigar ciertos problemas del NLP tradicional (ambigüedades, sinónimos, entre otros).

En la implementación de las técnicas, los análisis de NLP están basados en la plataforma UIMA [13]. UIMA es una arquitectura de última generación desarrollada por IBM que soporta el despliegue de diversos procesadores de texto para componer un análisis completo. Los analizadores textuales separan cada caso de uso en oraciones individuales y buscan identificar propiedades léxicas, sintácticas y semánticas del texto, como por ejemplo la estructura de predicado de una oración. En una primera etapa, los requerimientos de textos son analizados con técnicas estándares de NLP y enriquecidos con meta-información. Luego de este análisis inicial, cualquiera de las dos técnicas puede ser ejecutada para descubrir CCCs (ver Figura 1). La primera técnica, denominada CLUSTERING SEMÁNTICO, aplica un algoritmo de clustering que agrupa comportamientos semánticamente relacionados que se encuentran dispersos en múltiples documentos. Los agrupamientos son formados usando métricas que permiten calcular la similitud semántica entre palabras explorando las relaciones entre sinónimos y relaciones padre-hijo. La segunda técnica, denominada REGLAS ENRIQUECIDAS SEMÁNTICAMENTE, implementa un motor de búsqueda que permite definir consultas para concerns predefinidos usando información semántica de los

Figura 1: Esquema de las Técnicas de NLP Semánticas



casos de uso. Las consultas tienen como objetivo codificar conocimiento acerca de los concerns y como estos se relacionan semánticamente a las expresiones del lenguaje natural. Las siguientes sub-secciones describen los análisis de NLP inicial y las dos técnicas semánticas en detalle.

### III-A. Análisis de NLP Inicial

Un paso necesario para extraer conceptos de los requerimientos es reconocer las diferentes partes léxicas y sintácticas que constituyen el texto de los casos de uso. La información es extraída mediante anotadores de UIMA que encapsulan implementaciones de NLP de terceros. Se definieron una serie de anotadores que permiten recolectar datos acerca de las “oraciones”, “tokens” (es decir, palabras), como también sus propiedades (por ej., partes-del-discurso o “POS”, raíces léxicas o “stems”, raíces morfológicas o “lemmas”, palabras irrelevantes o “stopwords”, etc.). Adicionalmente, se incluyeron algoritmos para detectar relaciones sintácticas más complejas entre palabras, tales como “dependencias” y “pares de verbo y objetos directo”.

Los módulos de NLP generan capas de anotaciones para los casos de uso, es decir, meta-información de las oraciones en los documentos. Esta meta-información representa los resultados de los análisis de NLP y es almacenada para consultas futuras. Las implementaciones usadas en la mayoría de los módulos de NLP se encuentran disponible en la Web. Algunos de los paquetes más conocidos son: OpenNLP<sup>1</sup>, Stanford CoreNLP<sup>2</sup>, y Mate-Tools<sup>3</sup>.

### III-B. Clustering Semántico

Una forma de encontrar CCCs en requerimientos es buscar la presencia de verbos repetidos esparcidos en múltiples documentos [9]–[11]. Sin embargo, hay que tener en cuenta que existen sinónimos y vaguezas del lenguaje para evitar errores [15]. La primer técnica, denominada *Clustering Semántico*, combina un método de ML con semántica de las palabras para encontrar comportamientos similares repetidos. Específicamente, la técnica aplica un algoritmo de clustering jerárquico [1] que agrupa verbos que están semánticamente relacionados. Cada agrupamiento significativo encontrado se sugiere como un potencial CCC. La técnica se apoya en el inventario de sentidos WordNet<sup>4</sup> para recolectar información de las palabras [14] (por ej.,

su significado, sinónimos, antónimos, etc.) y métricas de similitud semántica que aprovechan la estructura de grafo de WordNet para cuantificar la relación entre las palabras y así poder agrupar aquellas más relacionadas [3].

A modo de ejemplo, considere el extracto de un caso de uso en la Figura 2. En esta oración, se describe un comportamiento representado por el verbo “sends”, posiblemente relacionado con el CCC de DISTRIBUTION. Con el clustering semántico, otras oraciones escritas con verbos semánticamente similares, tales como “transmits” o “transfers”, son agrupadas en un mismo cluster.

*III-B1. Determinación y Desambiguación del Significado de las Palabras:* La técnica utiliza WordNet para determinar el significado de las palabras. Este diccionario del lenguaje inglés contiene definiciones que explican y clarifican las palabras acorde a su contexto. En WordNet, los sustantivos y verbos son agrupados en conjuntos de palabras semánticamente relacionados llamados *synsets*, los cuales expresan conceptos particulares [14]. Cada *synset* incluye una definición y relaciones con otros *synsets* (sinónimos, antónimos, padre-hijo, entre otros). Estas relaciones interconectan los *synsets* armando una red de conceptos.

Desafortunadamente, las palabras usualmente son polisémicas y es necesario determinar su significado para cada contexto particular. En el área de NLP, la tarea para resolver este problema es denominada WSD (Word Sense Disambiguation). El WSD tiene el objetivo de desambiguar los significados de las palabras por medio de algoritmos computacionales [3]. Nuestra técnica utiliza el algoritmo de Lesk para desambiguar los casos de uso. Este algoritmo calcula el solapamiento de palabras entre las definiciones de los significados en el contexto de una oración.

*III-B2. Clustering Jerárquico utilizando Métricas de Similitud Semántica:* Por último, la técnica utiliza los significados de las palabras obtenidos anteriormente para agrupar comportamientos (es decir, verbos) en los casos de uso que están relacionados semánticamente. Con este fin, se utiliza un algoritmo de ML no supervisado denominado Clustering Jerárquico Aglomerativo HAC (Hierarchical Agglomerative Clustering) [1]. El algoritmo de clustering trata inicialmente cada palabra como un cluster individual de un solo elemento. Luego, va agrupando sucesivamente (o aglomerando) pares de clusters hasta que todos ellos han sido unidos en un único cluster que contiene a todas las palabras. Existen dos parámetros que afectan la salida del algoritmo HAC: la métrica de distancia y el método de linkage. El primer parámetro determina qué tan “cerca” está una palabra de otra. El segundo parámetro define la forma en que se determina la distancia entre diferentes clusters de palabras.

Para establecer la distancia entre las palabras, se decidió utilizar métricas semánticas que aprovechan la red de relaciones de WordNet para calcular su similitud. Estas métricas retornan valores entre 0 y 1, donde 0 indica que no hay similitud y 1 indica que hay mucha similitud entre las palabras. Existen varias métricas de similitud en la literatura [3]. En la técnica, se utilizaron dos de las métricas con mejores resultados<sup>5</sup>: *Lin* y *JCn*. Con respecto al método de linkage, se utilizaron tres métodos diferentes: *Nearest*

<sup>1</sup><http://opennlp.sourceforge.net/projects.html>

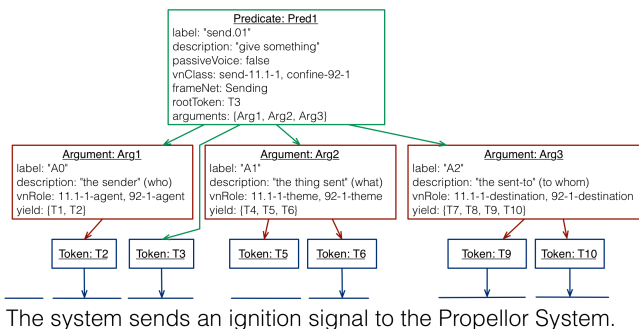
<sup>2</sup><http://nlp.stanford.edu/software/corenlp.shtml>

<sup>3</sup><http://code.google.com/p/mate-tools/>

<sup>4</sup><http://wordnet.princeton.edu>

<sup>5</sup>Las métricas de similitud semántica fueron adaptadas a métricas de distancia (compatibles con clustering) usando una transformación matemática.

Figura 2: Ejemplo del etiquetado con SRL



*Linkage, Furthest Linkage* y *Average Linkage*. Los dos primeros métodos solo observan los puntos en donde dos clusters se encuentran más cerca (o más lejos) uno del otro. El tercer método en cambio considera toda la estructura de los clusters para la comparación. Adicionalmente, para obtener un número plano y finito de clusters, la jerarquía retornada por el algoritmo fue cortada a un nivel dado de distancia semántica.

### III-C. Reglas enriquecidas Semánticamente

Una forma diferente de encontrar concerns en los requerimientos es definir reglas de búsqueda que permitan examinar las anotaciones para descubrir CCCs particulares [12]. La segunda técnica, denominada *Reglas enriquecidas Semánticamente*, aprovecha las anotaciones generadas por módulos semánticos de NLP y, mediante reglas de búsqueda, puede recuperar pistas de concerns individuales. Cada regla está compuesta por un conjunto de consultas que deben ser configuradas a priori por expertos (o modificadas por los analistas). Una consulta permite detectar partes de los casos uso que explícitamente mencionan un CCC específico analizando las anotaciones y sus propiedades. Además, se pueden usar consultas más avanzadas que permiten detectar menciones implícitas de un CCC en determinadas oraciones aprovechando anotaciones semánticas.

**III-C1. Análisis de Predicados y Argumentos:** Escribir reglas de búsqueda en términos de anotaciones de bajo nivel (por ejemplo, un POS) puede funcionar para buscar algunos concerns específicos, pero el desarrollo de reglas para concerns más complejos puede tornarse bastante difícil. Para simplificar la definición de las consultas, se incluyó un nuevo módulo que realiza un análisis de roles semánticos (Semantic Role Labeling, SRL). El SRL reconoce los predicados y argumentos de una oración, incorporando información del rol semántico que cada una de estas partes juegan en la oración (ver Figura 2). Con estas anotaciones, se espera que las consultas sean más amigables para los analistas (sin notación compleja de NLP) y tengan un mayor nivel de abstracción, pero que mantengan su precisión.

**III-C2. Especificación de Reglas Semánticas de Concerns:** Para la definición de reglas de búsquedas de concerns específicos definidos por los analistas, la técnica sigue una estrategia basada en reglas para identificar los CCCs y las oraciones afectadas por estos en los casos de uso. La idea es utilizar un lenguaje especializado para definir consultas de concerns las cuáles puedan recuperar información de las

múltiples capas de anotaciones generadas por los módulos NLP. Una consulta puede ser simple y trabajar con las anotaciones de bajo nivel de abstracción (por ej., tokens) para detectar un CCC explícito en el texto, o bien trabajar con anotaciones de mayor nivel de abstracción (por ej., predicados y argumentos) para detectar comportamientos que sutilmente pueden estar relacionados a un CCC particular. De esta forma, se complementan las búsquedas y se mejora la detección de CCC.

A modo de ejemplo, considere consultas para encontrar un concern de DISTRIBUTION. La consulta #1 (Q#1) especifica un patrón de búsqueda para encontrar anotaciones relacionadas al concern de DISTRIBUTION a través de los lemmas “broadcast”, “server”, and “channel”, entre otros. Q#1 es similar a los enfoques basados en palabras clave, pero incorpora anotaciones NLP como lemmas, stems y POS. La consulta #2 (Q#2) en cambio busca predicados cuya intención semántica indique la comunicación de algún tipo con una entidad (por ej., un sistema externo). Q#2 busca oraciones que incluyan al destinatario de algún mensaje analizando si la propiedad descriptiva de los argumentos es “send-to”. Muchos predicados, tales como “send.01”, “transfer.01” y “transmit.01” comparten este argumento semántico.

```

Q#1) SELECT S FROM SENTENCE AS S, TOKEN AS T
      WHERE FOR T(LEMMA = {'INTER-
                          NET', 'EXTERNAL', 'SERVER', 'CHANNEL', 'BROADCAST'})
      WHERE T.BEGIN > S.BEGIN AND T.END < S.END
Q#2) SELECT S FROM SENTENCE AS S, ARGUMENT AS A
      WHERE FOR A(DESCRIPTION = {'SENT-TO'})
      WHERE T.BEGIN > S.BEGIN AND T.END < S.END
    
```

Con el fin de ejecutar las consultas, se utilizó la tecnología provista por el proyecto EMF Query2<sup>6</sup> y así correr consultas similares a SQL sobre las anotaciones. Hasta el momento, se han definido reglas para concerns tales como PERFORMANCE, PERSISTENCE, PRESENTATION, INTEGRITY MANAGEMENT, EXTERNAL COMMUNICATION, USER ACCESS CONTROL, o ERROR MANAGEMENT, entre otras. Estas reglas fueron desarrolladas por analistas externos, los cuáles cuentan con experiencia en el análisis de requerimientos.

## IV. EVALUACIÓN

Para determinar la eficacia de las técnicas de NLP semánticas desarrolladas, se realizó una evaluación experimental con tres casos de estudio, analizando los CCCs encontrados en distintos documentos de requerimientos. Los casos de estudio provienen de diversos dominios, incluyendo: i) un sistema de gestión universitaria (CRS), ii) un sistema de mensajería institucional (HWS), y iii) un sistema de monitoreo de sensores (MSLite). Nuestra hipótesis es que las técnicas de NLP semánticas sirven para revelar requerimientos latentes, y funcionan mejor que las técnicas existentes.

Para evaluar las técnicas, se realizaron dos rondas de experimentos. En la primera ronda, se ejecutaron las técnicas de NLP semánticas (clustering y reglas) en los casos de estudio y se compararon los concerns detectados. Debido a que el clustering puede ser configurado de diferentes

<sup>6</sup><http://www.eclipse.org/modeling/emf/downloads/?project=query2>

formas, se probaron varias combinaciones de parámetros por caso de estudio, obteniendo un total de 18 resultados experimentales. Con respecto a las reglas semánticas, se probó la técnica en dos modos diferentes para investigar el rol de la información semántica en los resultados. Primero, se ejecutaron las reglas incluyendo sólo aquellas consultas que buscan sobre las propiedades de anotaciones de NLP simples, y luego se ejecutó de forma separada la técnica con aquellas consultas que buscan en anotaciones semánticas de SRL. Para la segunda ronda, se ejecutó una técnica provista por terceros y se analizaron sus resultados. Esta técnica esta implementada en EAMiner [10], una reconocida herramienta para procesar requerimientos y descubrir CCCs.

Para valorar el desempeño de las técnicas, se utilizaron dos métricas adaptadas de las disciplinas de IR y ML: *precision* y *recall* [1]. Lamentablemente, para obtener los valores de *precision* (y *recall*), es necesario que los resultados de las técnicas sean contrastados con una solución ideal (o golden standard). Es necesario contar con tres golden standards, uno para cada caso de estudio. Estos golden standards deben incluir los concerns y las relaciones crosscutting que se creen son correctas para el sistema analizado. Por esta razón, se le solicitó a dos analistas externos con conocimientos de CCCs que produzcan estos golden standards. En el caso de HWS y MSLite, los analistas contaron con análisis de concerns de alto nivel determinados por otros investigadores. Con la información de los golden standards, el cómputo se llevo a cabo de la siguiente forma. Una oración dentro de un caso de uso es considerada un verdadero positivo (tp) si el golden standard marca un CCC y la técnica efectivamente detectó ese CCC para esa oración. Se puede hacer un razonamiento similar para los verdaderos negativos (tn), falsos positivos (fp) y falsos negativos (fn). En este contexto, las interpretaciones de las métricas son las siguientes: *precision* mide cuantos de los CCCs detectados son correctos, mientras que el *recall* mide que tan completa la detección de CCC es, en el sentido de que ciertas oraciones afectadas por CCC pueden no haber sido tenidas en cuenta. Las funciones para calcular las métricas son las siguientes:  $precision = \frac{tp}{tp+fp}$  y  $recall = \frac{tp}{tp+fn}$ .

Los resultados de los experimentos, incluyendo las diferentes configuraciones de las técnicas de clustering semántico (SC) y las reglas enriquecidas semánticamente (SeR), como también los resultados de EAMiner, están reflejados en las figuras 3a, 3b and 3c (para CRS, HWS y MSLite, respectivamente). Para cada clustering, se definió una función de matching, en la cual cada cluster esta asociado a uno o más CCCs del golden standard. Se ejecutaron 6 instancias del clustering semántico para cada caso de estudio, variando dos parámetros: la métrica de distancia semántica y el método de linkage: Lin/Nearest, Lin/Furthest, Lin/Average, JCN/Nearest, JCN/Furthest, and JCN/Average.

El análisis de los valores de *precision* obtenidos por las configuraciones de clustering fueron similares para los tres casos de estudio (~50 %, ~60 % y ~70 %, para CRS, HWS y MSLite, respectivamente). Por otro lado, mientras que para el *recall* los resultados globales fueron bajos (~35 %), encontramos ciertas diferencias entre las configuraciones. Por ejemplo, las configuraciones Lin/Average y JCN obtuvieron el mejor *recall* en todos los casos de estudio. El

*recall* en MSLite mejoró lo observado en CRS y HWS (~30 %), obteniendo un ~40 % para casi todos los casos. En general, hubo algunas combinaciones de parámetros que superaron a otras.

Desde una perspectiva global, el método de linkage “Average” combinado con cualquiera de las métricas de distancia semántica (Lin y JCN) funciona mejor que las demás configuraciones. En comparación, la técnica de clustering semántico obtuvo una *precision* más baja que EAMiner en los tres casos de estudio (especialmente en CRS y MSLite), pero compensó esta disminución con un *recall* más alto. La baja *precision* fue porque muchos de los clusters sugeridos estaban relacionado semánticamente pero no tenían relación con los CCCs, aumentando el número de falsos positivos. Para el *recall*, el clustering semántico se comportó mejor que la técnica de EAMiner, obteniendo un incremento de *recall* de entre ~5 % (para HWS) y ~15 (para CRS y MSLite).

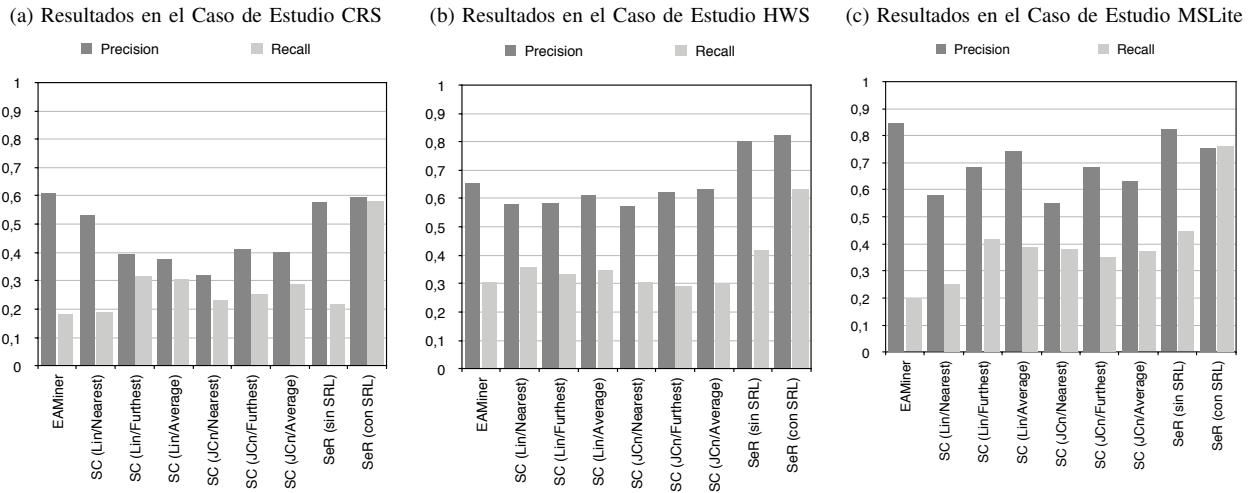
La *precision* de las reglas semánticas fue mejor que la de la técnica de clustering (en torno al ~20 % mejor). En comparación con EAMiner, las reglas obtuvieron una *precision* similar en CRS y MSLite (~60 % y ~80 %, respectivamente) y mejor en HWS (~65 % vs ~80 %). Cuando analizamos el *recall*, el desempeño de las reglas fue considerablemente superior al del clustering semántico y al de EAMiner (incluso cuando las consultas fueron definidas sin SRL). Además, cuando las reglas fueron complementadas con anotaciones de SRL, el *recall* aumento significativamente (~60 % para CRS y HWS y ~75 % para MSLite). La inclusión de anotaciones SRL en las consultas mejoró considerablemente los resultados de la técnica. En MSLite, por ejemplo, este incremento fue desde un ~45 % a un ~75 %. Dado que MSLite era el caso de estudio más grande y, por lo tanto, el escenario más demandante para minar concerns, estos resultados son alentadores para nuestra investigación.

## V. CONCLUSIÓN

La evolución de las técnicas de NLP ha hecho posible procesar documentos textuales y extraer información útil de forma precisa, hecho que tiene gran potencial para la industria. En este trabajo, presentamos dos técnicas basadas en NLP que aprovechan información semántica de los casos de uso textuales para revelar concerns latentes en las especificaciones de forma semiautomática. Para corroborar los beneficios de las técnicas propuestas, se llevó a cabo una serie de experimentos en tres casos de estudio y se evaluó su desempeño con las métricas de *precision* y *recall*. También se analizaron los resultados de una herramienta de terceros para apreciar cómo se comparan las técnicas de NLP semánticas con aquellas más simples. Los resultados obtenidos fueron prometedores. Ambas técnicas semánticas lograron un desempeño global bueno, mejorando los resultados de la técnica de terceros en términos de *recall*. Además, vale la pena mencionar que los resultados de las reglas fueron muy buenos, logrando una alta *precision* y un *recall* aún más alto.

La evaluación también reveló algunas limitaciones de nuestras técnicas. Primero, la calidad del texto de entrada a las técnicas afecta los resultados de los análisis de NLP. Segundo, con respecto a las reglas, la salida de la técnica puede ser afectada por la forma en que son escritas las consultas. Tercero, con respecto al clustering, varios de

Figura 3: Resultados Experimentales de las Técnicas (con distintas configuraciones)



los clusters detectados eran sugerencias incorrectas porque no tenían relación con los CCCs. Como trabajo futuro, se planea evaluar este tipo de técnicas con más casos de estudio. También se está analizando enriquecer las reglas y organizarlas para detectar concerns más específicos (por ej., concerns de diseño u objetivos de negocio). Finalmente, se están explorando mecanismos que puedan filtrar clusters de comportamientos irrelevantes y agrupar las palabras en base a conocimiento específico del dominio.

En resumen, creemos que las técnicas de NLP semánticas están listas para mejorar los procesos de producción ingenieril en la industria, y el desarrollo de herramientas a tal efecto es factible sin tener que realizar esfuerzos importantes.

#### RECONOCIMIENTOS

Este trabajo está financiado en forma parcial por ANPCyT (Argentina) a través del proyecto PICT 2010 No. 2247.

#### REFERENCIAS

- [1] R. Baeza-Yates, B. Ribeiro-Neto *et al.*, *Modern information retrieval*. ACM press New York., 1999, vol. 463.
- [2] T. Mitchell, *Machine Learning*, ser. McGraw-Hill series in Computer Science. McGraw-Hill, 1997.
- [3] R. Navigli, "Word sense disambiguation: A survey," *ACM Comput. Surv.*, vol. 41, no. 2, pp. 10:1–10:69, 2009.
- [4] M. Palmer, D. Gildea, and N. Xue, *Semantic Role Labeling*, ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool, 2010.
- [5] L. Mich, M. Franch, and P. L. Novi I., "Market research for requirements analysis using linguistic tools," *Requirements Engineering*, vol. 9, no. 2, p. 151, 2004.
- [6] B. Sateli, E. Angius, S. S. Rajivelu, and R. Witte, "Can text mining assistants help to improve requirements specifications?" in *Mining Unstructured Data (MUD 2012)*, Canada, 2012.
- [7] A. Rago, C. Marcos, and A. Diaz-Pace, "Uncovering quality-attribute concerns in use case specifications via early aspect mining," *Requirements Engineering*, vol. 18, no. 1, pp. 67–84, 2013.
- [8] J. Araujo, E. Baniassad, P. Clements, A. Moreira, A. Rashid, and B. Tekinerdogan, "Early aspects: The current landscape," *Technical Notes, CMU/SEI and Lancaster University*, 2005.
- [9] E. Baniassad, P. Clements, J. Araujo, A. Moreira, A. Rashid, and B. Tekinerdogan, "Discovering early aspects," *Software, IEEE*, vol. 23, no. 1, pp. 61–70, 2006.
- [10] A. Sampaio, A. Rashid, R. Chitchyan, and P. Rayson, "Ea-miner: towards automation in aspect-oriented requirements engineering," *Transactions on AOSD III*, pp. 4–39, 2007.
- [11] B. Ali and Z. Kasirun, "3ci: A tool for crosscutting concern identification," in *Computational Intelligence for Modelling Control & Automation*. IEEE, 2008, pp. 351–355.
- [12] L. Rosenhainer, "Identifying crosscutting concerns in requirements specifications," in *Proc. of OOPSLA '04*, 2004.
- [13] D. Ferrucci and A. Lally, "UIMA: an architectural approach to unstructured information processing in the corporate research environment," *Natural Language Engineering*, pp. 327–348, 2004.
- [14] C. Fellbaum, *WordNet: An Electronic Lexical Database*, ser. Language, Speech, and Communication. MIT Press, 1998.
- [15] A. Rago, E. Abait, C. Marcos, and A. Diaz-Pace, "Early aspect identification from use cases using NLP and WSD techniques," in *Workshop on Early Aspects held at AOSD'09*, 2009, pp. 19–24.