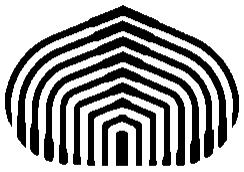


UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

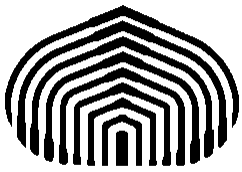
EXTRACCIÓN DE INFORMACIÓN FOCALIZADA BASADA EN RESPUESTAS INCOMPLETAS (FIE)

Por:
FRANCISCO RODRÍGUEZ DRUMOND

Realizado con la asesoría de:
PROF. SORAYA ABAD

PROYECTO DE GRADO
Presentado ante la Ilustre Universidad Simón Bolívar
como requisito parcial para optar al título de
Ingeniero de Computación

Sartenejas, Septiembre de 2012



UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

ACTA FINAL PROYECTO DE GRADO

EXTRACCIÓN DE INFORMACIÓN FOCALIZADA BASADA EN RESPUESTAS INCOMPLETAS (FIE)

Presentado por:
FRANCISCO RODRÍGUEZ DRUMOND

Este Proyecto de Grado ha sido aprobado por el siguiente jurado examinador:

_____
PROF. 1

_____
PROF. 2

_____
PROF. 3

Sartenejas, X/X/12

Resumen

RESUMEN GOES HERE - To be done.

Acrónimos

FIE	Extracción Focalizada de Información	Focalized Information Extraction
DIA	Arquitectura de Interrogación de Documentos	Document Interrogation Architecture
ODIL	Lenguage de interrogación de documentos con ontologías	Ontology Document Interrogation Language
BFR	Buscador Focalizado de Información	Buscador Focalizado de Información
DOI	Determinador del Origen de Incompletitud	Determinador del Origen de Incompletitud
EFI	Extractor Focalizado de Información	Extractor Focalizado de Información
NLP	Procesamiento del Lenguaje Natural	Natural Language Processing
BIE	Extracción Amplia de Información	Broad Information Extraction
NCF	Forma Normal Conjuntiva	Normal Conjunctive Form

DUDA: Que hago con los acrónimos que son sólo en español? Deberia poner los acronimos siempre en inglés?

Índice general

Índice general vi

Índice de tablas viii

Índice de figuras x

Introducción 1

1 Marco Teórico 3
1.1 Revisión de trabajos sobre consultas con datos incompletos. 3
1.2 DIA y el lenguaje ODIL 5

2 Análisis de incompletitudes 8
2.1 Mecanismo para encontrar qué valores de atributos faltan en una consulta. 9
2.2 Forma canónica de los diferentes tipos de consulta. 12
2.2.1 Consulta simple 12
2.2.2 Consulta con funciones agregadas 13
2.2.3 Consultas anidadas. 15
2.2.4 Consultas con agregados 17
2.2.5 Consultas con relaciones de especialización-generalización. 18

3 Diseño 19
3.1 El Determinador del Origen de Incompletitud (DOI) 21
3.1.1 El Contexto de Extracción 22
3.2 El Buscador Focalizado de Información (BFI) 24

4 Detalles de implementación 27
4.1 Consideraciones generales 27
4.1.1 Lenguaje de programación 27
4.1.2 Librerías utilizadas 27
4.2 Implementación del Determinador del Origen de Incompletitud (DOI) 28

4.2.1 Transformación de las condiciones a la Forma Normal Conjuntiva 28
4.2.2 Implementación del constructor de contextos de extracción 28
4.3 Implementación del Buscador Focalizado de Información (BFI) 29
4.3.1 Preprocesador de Textos 29
4.3.2 Extractor Focalizado 30
4.3.3 Diseño de las expresiones regulares para la configuración del Preprocesador de textos y del Extractor Focalizado 31

5 Pruebas **34**
5.1 Consideraciones Generales 34
5.2 Pruebas unitarias sobre el Determinador de Origen de Incompletitud. 35
5.3 Pruebas unitarias sobre el Buscador Focalizado de Información 35
5.3.1 Pruebas del Preprocesador de Texto 35
5.3.2 Pruebas de Extractor Focalizado 35

6 Resultados **37**
6.1 Resultados de las pruebas unitarias sobre el Módulo de Determinación de Origen de la Incompletitud. 37
6.2 Resultados de las pruebas unitarias sobre el Módulo de Búsqueda Focalizada de Información 37
6.2.1 Resultados de las pruebas unitarias del Preprocesador de Textos 37
6.2.2 Resultados de las pruebas unitarias del Extractor Focalizado 37
6.2.3 Análisis de los resultados de las pruebas unitarias del Buscador Focalizado de Información 38

Conclusiones y recomendaciones **46**

Bibliografía **48**

Apéndices **50**

A Ejemplos de archivos de configuración XML de Preprocesador de Texto y Extractor Focalizado **51**

B Resultados de las pruebas unitarias realizadas sobre el Buscador Focalizado de Información **54**

Índice de tablas

6.1 Resultados detallados la evaluación del Preprocesador de Textos 37
6.2 Resultados de la evaluación del Extractor Focalizado - Dominio: Designaciones. UnitHit Measure mínimo:1.0 38
6.3 Resultados de la evaluación del Extractor Focalizado - Dominio: Designaciones. UnitHit Measure mínimo:0.66 39
6.4 Resultados de la evaluación del Extractor Focalizado - Dominio: Designaciones. UnitHit Measure mínimo:0.33 39
6.5 Resultados de la evaluación del Extractor Focalizado - Dominio: Escalafon. UnitHit Measure mínimo:0.66 40
6.6 Resultados de la evaluación del Extractor Focalizado - Dominio: JuradosAscenso. UnitHit Measure mínimo:0.66 40

B.1 Resultados de la evaluación del Extractor Focalizado - Dominio: Designaciones. UnitHit Measure mínimo:1.0 55
B.2 Resultados de la evaluación del Extractor Focalizado - Dominio: Designaciones. UnitHit Measure mínimo:0.66 56
B.3 Resultados de la evaluación del Extractor Focalizado - Dominio: Designaciones. UnitHit Measure mínimo:0.33 57
B.4 Resultados de la evaluación del Extractor Focalizado - Dominio: Escalafon. UnitHit Measure mínimo:1.0 58
B.5 Resultados de la evaluación del Extractor Focalizado - Dominio: Escalafon. UnitHit Measure mínimo:0.66 59
B.6 Resultados de la evaluación del Extractor Focalizado - Dominio: Escalafon. UnitHit Measure mínimo:0.33 60
B.7 Resultados de la evaluación del Extractor Focalizado - Dominio: JuradosAscenso. UnitHit Measure mínimo:1.0 61
B.8 Resultados de la evaluación del Extractor Focalizado - Dominio: JuradosAscenso. UnitHit Measure mínimo:0.66 62

B.9 Resultados de la evaluación del Extractor Focalizado - Dominio: JuradosAscenso. UnitHit Measure mínimo:0.33 63

Índice de figuras

1.1 Arquitectura de Interrogación de documentos 6

3.1 Diagrama del Buscador Focalizado de Información 25

4.1 Funcionamiento interno del Buscador Focalizado de Información 33

Introducción

Uno de los retos aún presentes en el mundo de la computación hoy por hoy es el de construir herramientas que permitan realizar operaciones de procesamiento de un lenguaje natural. Se entiende por lenguaje natural un idioma como el inglés, español, italiano, que no es reconocible por una computadora sin herramientas pertientes. Cada día surgen más y más tecnologías en este ámbito: existen procesadores de comandos de voz que son capaces de entender instrucciones dadas por un usuario, programas que permiten tomar dictados de un humano, traductores de textos y programas que colocan signos de puntuación automáticamente, entre otros. A la par de dichos avances, sin embargo, hay muchísimos campos los cuales se hacen más difíciles para la computación; en la medida en la que los objetivos trazados requieren una mayor "comprensión" semántica del lenguaje natural se hace más complicado construir herramientas que resuelvan los problemas planteados.

Uno de las áreas del Procesamiento de Lenguaje Natural (NLP) que se está desarrollando actualmente y cuya utilidad es altísima es la extracción de información (IE, que significa Information Extraction). La IE tiene como objetivo ubicar dentro de un texto datos que puedan ser útiles para responder preguntas sobre el dominio del texto. (o en general datos en cualquier medio que se encuentre en algún lenguaje humano). Por ejemplo, puede ser deseable construir un extractor que analice noticias de un periódico para poblar una base de datos con los principales eventos que han ocurrido. Dicha base de datos puede luego ser consultada de forma que un usuario pueda buscar algún dato que sea necesario sin tener que leer todas las noticias. Generalizando, los avances en esta área pueden ser aplicados en cualquier dominio en el cual se tenga una amplia cantidad de información en lenguaje natural no entendible por una máquina: ámbitos corporativos, educativos, diplomáticos, etcétera. Las posibles aplicaciones de ésto son numerosas.

Abad Mota (2009) propuso en [1] una arquitectura para la Interrogación de Documentos (DIA, por sus siglas Document Interrogation Architecture) que busca resolver el problema de extracción de información en documentos escritos. La interrogación de documentos se entiende como un proceso en el cual tienen lugar varias actividades: una primera extracción de información de un conjunto de documentos, la realización de consultas sobre una ontología poblada con información previa, y la resolución de consultas con información que no se encuentra presente en la ontología o en la base de datos, accediendo a los

documentos.

Ruiz en [2] y [3] propuso mecanismos para realizar la primera extracción definida en DIA para poblar inicialmente la ontología. Apaza y Mirisola en [4] e Ituarte en [5] implementaron el lenguaje ODIL (Lenguage de interrogación de documentos con ontologías) diseñado a la par de DIA para la interrogación de Documentos. Todos estos avances están relacionados con las primeras fases propuestas en DIA.

El propósito del presente proyecto de grado es diseñar e implementar un mecanismo de *extracción focalizada*, que permita resolver consultas cuando la ontología y la base de datos poblada no se encuentren completos. Esto es, el objetivo del mismo es construir un mecanismo que permita mediante búsquedas enfocadas encontrar datos faltantes en la ontología a partir de un conjunto de documentos e información presente en la ontología. Esta fase es la última propuesta en DIA y opera bajo la premisa de que es posible obtener mejores resultados en IE cuando se tiene información conocida.

Para poder realizar la extracción focalizada es necesario una serie de pasos que serán descritos en profundidad en este informe. Por ahora conviene mencionar como objetivos específicos o pasos intermedios: el razonamiento sobre el problema de las respuestas incompletas o imprecisas a consultas a la base de datos a partir del lenguaje ODIL, el razonamiento sobre cómo aprovechar la información presente en la ontología para construir un extractor de información y el diseño e implementación de un extractor focalizado.

El presente informe busca sintetizar los principales resultados obtenidos en este proyecto. Para ello está estructurado en 7 capítulos. Primeramente se hace un planteamiento del problema introduciendo más en detalle la arquitectura DIA y presentando en profundidad el concepto de extracción focalizada. Seguidamente se presenta una breve revisión de algunos trabajos existentes en el ámbito de extracción de información. Posteriormente se presentan en los capítulos 3 y 4 el diseño propuesto para la extracción focalizada de la información y el mecanismo concebido para determinar los origenes de incompletitud propios de una ontología incompleta. Luego se realizan algunos comentarios relevantes sobre la implementación del sistema, se presentan las pruebas realizadas para probar el sistema en conjunto con sus resultados y el análisis pertinente.

Capítulo 1

Marco Teórico

A continuación se resume brevemente algunos trabajos de interés. Posteriormente se introducen algunos conceptos relacionados con DIA y el lenguaje utilizado para interrogar documentos, ODIL, que son necesarios para el entendimiento de este proyecto.

1.1. Revisión de trabajos sobre consultas con datos incompletos.

En "A survey of Uncertain Data Algorithms and Applications" (2005) en [6], Aggarwal y Fellow presentan un estudio de las tecnologías que se han desarrollado para trabajar con datos inciertos. Los autores definen los datos inciertos como datos que pueden estar incompletos o que pueden contener errores. Por ejemplo, se puede tener una base de datos con datos de mediciones meteorológicas cuya precisión depende de los instrumentos utilizados y que por ende sus datos pueden contener errores. Puede darse también el caso en el que no se tenga la información completa, y que haya la presencia de varios atributos nulls en la base de datos. En el caso particular de este proyecto de grado se trabaja con datos incompletos.

En base a la definición de datos inciertos, los autores examinan las técnicas más recientes en tres campos fundamentales: modelado de datos inciertos, manejo de datos inciertos y minería de datos inciertos. Este proyecto de grado está enmarcado en las dos primeras categorías: modelado y manejo de datos inciertos. Esto se debe a que se quiere estudiar la forma como modelar el problema así como un conjunto de mecanismos que permitan completar la información que falta mediante extracción focalizada.

Los autores definen una base de datos probabilística como un espacio de probabilidad finito en el cual los resultados son las posibles bases de datos consistentes con un esquema dado. En pocas palabras, se tiene una representación de los "posibles mundos". Existen varias soluciones para ello, como modelar la probabilidad de que una tupla esté en la base de datos o, si se quiere más detalle, la probabilidad de que un atributo de una tupla de base de datos esté presente. Sobre esta base se pueden construir técnicas para

el manejo de datos y para realizar minería sobre ellos.

Por otro lado, Aggarwal y Fellow (2006) en [6] proponen utilizar una función de probabilidad sobre la correctitud de los datos presentes en la base de datos. Si bien el objetivo de este proyecto es el desarrollo de un mecanismo para contestar una consulta cuya respuesta no está en su totalidad en la ontología, la aproximación que hacen estos autores puede ser útil para definir una medida de la calidad de los datos presentes en la base de datos.

Es importante destacar sin embargo, una referencia que se hace al trabajo "Evaluating Probabilistic Queries over Imprecise Data" (2003), de Cheng, Kalashnikov y Prabhakar en [7]. En el mismo se hace un análisis sobre las consultas con datos imprecisos o propensos a errores y proponen una clasificación de consultas probabilísticas que puede ser tomada en cuenta para clasificar las consultas con respuestas incompletas. Básicamente los autores toman en consideración dos criterios: el tipo de elemento devuelto por la consulta y el uso de funciones agregadas. En pocas palabras, cuando se toma en consideración el tipo de elemento devuelto por la consulta se tiene que puede devolver un valor puntual o un conjunto de tuplas. En el contexto de las consultas probabilísticas esto puede tener implicaciones: para las consultas que buscan un valor los autores definen cotas superiores e inferiores que definen intervalos en los que los valores de la función deben estar, con una probabilidad acumulada de 1. La segunda clasificación toma en cuenta si existen agregados o no. La presencia de agregados puede afectar la evaluación de consultas incompletas.

Además de estos dos trabajos se han examinado otros tres. Sin embargo, su aporte y utilidad para el presente de trabajo de investigación es menor. Chen, Chen y Xie proponen en "Cleaning Uncertain Data with Quality Guarentees" (2008) [8], una métrica para cuantificar la ambigüedad de una respuesta de consulta bajo semánticas de mundos posibles. Sobre esta base, se podría construir un mecanismo para "limpiar" la base de datos.

El trabajo "On databases with Incomplete Information" de Lipski (1981) [9] es muy citado por otros investigadores en el área de consultas incerteras y sin duda alguna constituye un hito muy importante en ésta area. Sin embargo, en dicho trabajo se busca tratar el tema con un formalismo matemático que va más allá de los alcances de este proyecto.

Por otro lado, Kang y Kim proponen en "Query type classification for web document retrieval" (2003) [10] un mecanismo de clasificación de consultas para extracción de información del WEB. Dicha clasificación corresponde con el tipo de operación que se desea: úbicar algo por tópico, ubicar un homepage o ubicar un servicio.

Por último, Rocquenco, Segoufin y Viano presentan en "Representing and querying XML with incomplete information" (2001) [11] un modelo para hacer consultas incompletas sobre XML. Dicho trabajo no fue de utilidad dado que la naturaleza de las mismas es diferente de las que se hacen en el contexto de DIA, basadas en una ontología y en un modelo de datos más formal que el de un XML.

1.2. DIA y el lenguaje ODIL

DIA es una arquitectura que permite realizar consultas sobre un conjunto de documentos utilizando una representación basada en ontologías por medio de un lenguaje llamado *ODIL* (Ontology-based Document Interrogation Language). Una ontología, en un contexto computacional, es una especificación basada en primitivas con los cuales se quiere modelar un dominio de conocimiento o del discurso (Tom Gruber 2009 [12]). Dichas primitivas a menudo son clases o entidades, atributos y relaciones. Una ontología puede ser vista consecuentemente como una abstracción en un modelos de datos cuyo propósito es modelar conocimiento (Ibidem).Para hacer consultas DIA funciona en 3 fases: preparación de datos, extracción de datos e interrogación de documento.

La primera fase de DIA consiste en la elección de un conjunto de documentos sobre el cual se realizará la interrogación y en la construcción de un extractor de información. El extractor de información puede realizarse utilizando mecanismos de procesamiento de lenguaje natural, de aprendizaje de máquina, estadísticos, entre otros posibles.

La segunda fase consiste en realizar una primera Extracción Amplia de Información *(BIE)* del conjunto de documentos utilizando el extractor ya construido par poblar una Base de Datos relacional.

En la tercera fase los usuarios, toda vez que se haya poblado la base de datos con datos, pueden realizar preguntas en la base de datos de información extraida de los documentos. En este punto se debe determinar la respuesta basado en la información presente en la Base de Datos. Para ello es necesario revisar si hay algún tipo de incompletitud y en el caso de haberla, buscar de nuevo en el conjunto de documentos. El proceso de realizar una segunda extracción sobre documentos es el denominado Extracción Focalizada de Información. En esencia busca encontrar la información faltante para poblar la base de datos para luego responder la pregunta realizada. La figura 1.2 sintetiza las fases de DIA explicadas

El objetivo de la Extracción Focalizada de Información es encontrar información desconocida en la ontología realizando una extracción sobre un conjunto de documentos utilizando información presente en la ontología. El conjunto de documentos sobre el cual se hace la segunda extracción puede ser el mismo sobre el cual se hizo Extracción Amplia de Información o no. Una de las ventajas de esto es que se puede

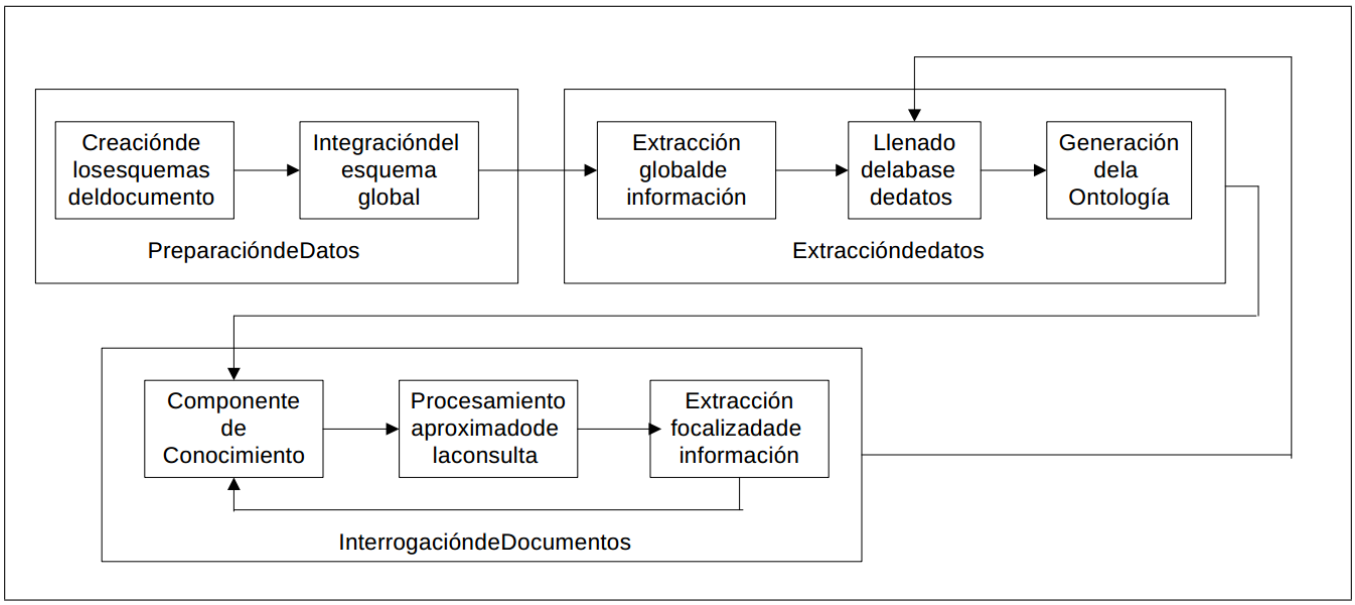


Figura 1.1: Arquitectura de Interrogación de documentos

explorar un conjunto de documentos de mayor tamaño al inicial, incluyendo por ejemplo la *WWW* (World Wide Web), sobre el cual es más complejo realizar una extracción inicial debido a su magnitud. La utilización de información conocida puede ser explotada por el extractor para simplificar considerablemente la búsqueda de la información faltante.

Para poder realizar la interrogación de documentos en DIA, se diseñó e implementó el lenguaje ODIL. ODIL, que significa Ontology-based Document Interrogation Language (Lenguaje basado en ontología para la interrogación de documentos), fue definido por Abad y Helman en [13] para realizar preguntas sobre los documentos en la arquitectura DIA. Para ello, ODIL cuenta con dos módulos diferentes: el Módulo de Definición de Conceptos y el Módulo de Cláusulas de Interrogación. El primer modulo busca definir la ontología que se utilizará para hacer preguntas sobre los documentos. El segundo tiene como objetivo permitir al usuario hacer las preguntas.

El eje central de ODIL es el *concepto*. Un concepto se refiere a un elemento del dominio de información sobre el cual se define la ontología. En lo sucesivo se utilizará la expresión *dominio de información* para referirse al ámbito o temática de los documentos sobre los cuales se hace la extracción. El concepto debe tener un nombre, una definición basada en cinco tipo de abstracciones: concepto primitivo, agregación, enumeración y generalización, un origen y un conjunto de sinónimos.

La abstracción se refiere a la tipología del concepto. Un concepto primitivo puede verse como un átomo de información (un pedazo de información que no se puede descomponer), como por ejemplo un atributo

del modelo ER. Un concepto puede verse también como un agregado de otros conceptos, lo cual permite darle estructura al modelo de datos. Por ejemplo, manteniendo el paralelismo con el modelo ER, un concepto puede verse como una entidad que a su vez está definida por un conjunto de atributos. Además de la agregación, ODIL permite establecer jerarquías de especialización-generalización, similar al modelo OMT. Por último, la abstracción enumeración se refiere a la especificación de instancias de un concepto.

Otro elemento importante para entender ODIL es que el concepto puede tener dos orígenes diferentes: puede estar basado en información contenida en la Base de Datos o puede estar definido exclusivamente a nivel de la ontología. Si se encuentra en la base de datos, un concepto puede verse como algún elemento del modelo ER, necesario para la persistencia de los datos. Se puede suponer que la base de datos tiene un modelo relacional ya que ODIL y DIA operan bajo esta suposición. Un concepto con origen en la base de datos puede ser, por ejemplo, un atributo o una entidad.

Por otro lado, un concepto definido a nivel de la ontología tiene como objetivo, en lugar de permitir la persistencia de datos, permitir a un usuario realizar preguntas. Por ejemplo, supongamos que se tiene una ontología sobre estudiantes de la Universidad Simón Bolívar. Un concepto con origen en la base de datos puede ser un carnet (concepto primitivo), puede ser un estudiante (agregación de varios conceptos), o puede ser un estudiante de postgrado (especialización-generalización). Un concepto definido en la ontología puede ser un equipo de trabajo para algún proyecto, que a su vez es una agregación sobre conceptos (los estudiantes) con origen en la Base de Datos. Al hacer consultas, el usuario puede definir la estructura de qué es un equipo de trabajo y puede definir instancias de equipos de trabajos basadas en instancias de estudiantes contenidas en la Base de Datos.

Entender que pueden haber conceptos definidos a nivel de la ontología es importante para razonar sobre los orígenes de incompletitud. Puede haber información en los documentos sobre los cuales se hizo la primera extracción de DIA que no fue extraída porque en el momento en el cual se hizo dicha primera extracción el modelo no contemplaba dichos conceptos.

Por último es de interés para el lector entender algunos aspectos sobre la implementación del lenguaje ODIL. Dicha implementación fue hecha por Apaza y Mirasola en [4] y consiste en un sistema de software que analiza texto escrito con la sintaxis de ODIL para definir estructura y hacer consultas y basado en el mismo, contesta preguntas con el contenido de la Base de Datos. Lo relevante para la comprensión de este proyecto de grado es saber que dichas consultas son respondidas traduciendo la consulta hecha en ODIL a expresiones del algebra relacional. Es útil saber esto ya que como se verá en los próximos capítulos, para descubrir los orígenes de incompletitud de las consultas es necesario razonar sobre la consulta propiamente dicha, y es conveniente suponer que esa consulta se realiza utilizando álgebra relacional.

Capítulo 2

Análisis de incompletitudes

La incompletitud en la respuesta de una consulta puede estar dada por varios motivos. Puede ocurrir que falta una tupla en la base de datos, que se desconoce una relación entre dos tuplas o por el desconocimiento del valor de uno o más atributos de una tupla.

El primer caso tiene quizás la mayor dificultad a la hora de determinar con precisión que existe incompletitud ya que no hay una forma de saber, a partir de los datos que se tienen cargados en la base de datos y la estructura de la ontología, que falta una instancia. El segundo caso también es complejo por cuanto se necesita poder determinar si dos instancias están relacionadas siguiendo la semántica de una interrelación que se encuentre en el modelo ER o en la ontología en base al texto. El tercer caso es el menos complejo de los tres pues se puede trabajar con la información que ya se tiene para determinar qué atributos faltan y posteriormente realizar la búsqueda focalizada, usando potencialmente pistas o datos ya contenidos en la ontología o la BD o datos de la consulta misma.

En este proyecto se trabajará sólo con el tercer caso, la determinación de valores de atributos faltantes para poder dar respuesta a una consulta.

En este capítulo se presenta un resumen de los resultados obtenidos para determinar qué atributos tienen valores nulos necesarios para responder una consulta. Para ello se presenta primeramente un análisis y una presentación general del mecanismo para encontrar qué valores de atributos faltan en una consulta. En una segunda instancia se presenta una clasificación de las consultas según las operaciones que se deben hacer para determinar los origenes de la incompletitud y el mecanismo para determinar los origenes de incompletitud particular para cada una de ellos.

2.1. Mecanismo para encontrar qué valores de atributos faltan en una consulta.

Es conveniente empezar diciendo que la resolución de una consulta de ODIL pasa por tres pasos. Primero, la consulta de ODIL es traducida a una o más consultas SQL que se ejecutan en una Base de Datos Relacional. Dependiendo de la estructura de la ontología (la presencia de agregaciones o generalización-especialización), se hace necesario componer la respuestas a las consultas SQL ejecutadas para dar una respuesta a la consulta original en ODIL. Por último, se evalúa esta respuesta para determinar si está incompleta o no.

En este contexto, el proceso de encontrar qué valores de atributos faltan en una consulta consiste en esencia en determinar para cada una de las instancias de los conceptos que participan en una consulta, qué atributos tienen valores nulos que son necesarios para responder a la consulta. Podría darse el caso de que por decisiones de diseño de la ontología o de la base de datos sea posible asignar un valor por defecto, diferente de nulo, para señalar incompletitud; en caso de que esto ocurra la tarea de encontrar nulos se reemplaza por la tarea de inspeccionar los valores de los atributos participantes en la consulta para encontrar el valor por defecto especificado.

Para realizar la verificación mencionada, se pueden hacer consultas sobre cada uno de los conceptos participantes en una consulta en las que aparezcan los atributos listados en la consulta, en el LIST, SUCH THAT o en una consulta anidada. Dichas consultas pueden preguntar qué atributos de los necesarios para responder a la consulta tienen valores nulos.

Para cumplir con el requerimiento de que se evalúen todas las instancias presentes en la Base de Datos y para poder identificar las instancias con datos faltantes, se puede utilizar la noción de identificador de un concepto en la ontología, similar al de clave primaria. Si bien el concepto de clave primaria está relacionado con el Modelo Relacional, puede perfectamente adaptarse en el contexto de ontologías. En tal sentido, una clave primaria (o identificador) en una ontología puede verse como un conjunto de propiedades o atributos tales que juntos identifican unívocamente una instancia.

Es conveniente recordar que la estructura general de una consulta en ODIL está dada por dos partes. En primera parte se tiene la claúsula LIST INSTANCES, en donde se colocan los atributos que son necesarios como respuesta a la consulta. En segunda instancia se tiene la cláusula SUCH THAT en la que se realiza la calificación de la consulta. Esto implica que para cada consulta en la que se quiera verificar incompletitud se deban hacer como mínimo dos conjuntos de consultas. Sean de aquí en adelante L1 la lista de los atributos que forman parte de la respuesta de la consulta y L2 la lista de los atributos que califican la consulta. El mecanismo para encontrar qué atributos tienen valores faltantes consiste en

realizar dos tipos de consultas que se definirán a continuación.

El primer tipo de consulta sirve para determinar un tipo de incompletitud de la Base de Datos en el cual no se puede responder la consulta del usuario porque la información que se desea no está presente. Dado que se quiere realizar extracción focalizada de información para completar los valores faltantes, es pertinente utilizar la calificación completa de la consulta original, con todas las condiciones que usan atributos de L2 diferentes del atributo sobre el cual se realiza la extracción focalizada. Esto es, es deseable que en las consultas adicionales para encontrar valores de un atributo que falta se mantengan las condiciones de los otros atributos pertenecientes a L2. De lo contrario se trataría de una búsqueda de atributos incompletos que se puede realizar en cualquier momento, y como no se usaría información ya conocida en la ontología sería un proceso diferente de lo que DIA define como extracción focalizada.

El segundo tipo de consulta adicional está ligado a otra forma de incompletitud, en el cual es posible que la respuesta no esté completa porque no es posible encontrar qué instancias cumplen con las condiciones del SUCH THAT. Este tipo de consulta se realiza para cada uno de los atributos que participa en la calificación de la consulta, utilizando el resto de las condiciones del SUCH THAT que no incluyen al atributo que se esté examinando. Al utilizar el resto de las condiciones sin incluir a uno de los atributos se está realizando una calificación parcial de la consulta. Esto se debe a que al igual que en el primer tipo de consulta, se quiere realizar extracción focalizada y por lo tanto interesa que haya una calificación sobre los atributos siempre y cuando no sea sobre el atributo sobre el que se quieren buscar valores nulos. Esto último se debe a que de lo contrario podrían no encontrarse todos los valores nulos.

Para realizar la calificación parcial de la consulta, es necesario eliminar exactamente las partes de la condición en las cuales se encuentra el atributo en cuestión. Dado que la cláusula SUCH THAT puede estar formada por una serie de conjunciones y disjunciones de condiciones, es necesario elegir bien que partes de la condición se quieren eliminar. Se busca eliminar de la consulta todas las ocurrencias del atributo en la calificación manteniendo el mayor número de calificaciones sobre otros atributos.

Para hacer esto, se puede suponer que el SUCH THAT se encuentra en una Forma Normal Conjuntiva (NCF) - esto es, está formado por una secuencia de conjunciones de subcondiciones formadas exclusivamente por disjunciones de átomos (Weisstein, sin fecha [14]). En el caso en el que el SUCH THAT no se encuentre en la NCF, es posible realizar una serie de transformaciones sobre la mismas aplicando tantas veces como sea necesarios los teoremas de Morgan y otras propiedades lógicas.

Una vez que se tiene la NCF, se deben eliminar las cláusulas que precisamente conciernen al atributo que se está inspeccionando. La necesidad de utilizar la FNC es que si se tiene una cláusula que involucre al atributo A en cuestión en disjunción con condiciones sobre una lista de otros atributos LA, eliminar la

condición sobre A puede hacer más restrictiva la calificación hecha en el SUCH THAT.

Por ejemplo, si se tiene la siguiente consulta:

LISTA estudiante.carnet INSTANCES
SUCH THAT estudiante.carrera = 'Ingeniería de Computación' OR estudiante.indice <3

Con los siguientes datos:

Carnet	Carrera	Indice
06-40000	Ingeniería de Computación	2
06-40001	null	2
06-40002	Ingeniería de Computación	4
06-40003	null	4

Entonces hacer la consulta:

LISTA estudiante.carnet INSTANCES SUCH THAT (estudiante.carrera is null) AND estudiante.indice <3; (en la que se eliminó la condición sobre estudiante.carrera del or)

Devuelve como resultado:

06-40001

Mientras que se esperaría que la respuesta sea

06-40001
06-40003

Que son todas las tuplas que potencialmente podrían responder a la consulta realizada (puede darse que el estudiante con carnet 06-40003 estudie computación y por lo tanto deba aparecer en el resultado de la consulta realizada por el usuario aunque su indice sea mayor o igual a 3).

Habiendo aclarado cómo se realiza la calificación parcial, es pertinente realizar dos comentarios adicionales sobre estos tipos de consulta explicados. Tiene sentido realizar todas las consultas que buscan hallar valores de atributos faltantes en la SUCH THAT antes de buscar valores de atributos que se encuentran en la LIST. El motivo de esto es que es necesario garantizar que exista el menor número de valores nulos para atributos usados para calificar la consulta 1, de forma que se disminuya el riesgo de que la respuesta esté incompleta porque no se encuentren todas las tuplas que califiquen en la consulta.

En segundo lugar, el proceso de extracción focalizada puede ser de tipo iterativo y que la búsqueda focalizada puede repetirse hasta que se cumpla que no se encuentren más valores de atributos. El motivo de esto es que si en la búsqueda focalizada se utiliza información sobre los atributos con valores diferentes de nulos de las instancias que están incompletas para facilitar la ubicación de los atributos faltantes, entonces la posibilidad de encontrar valores de atributos aumenta a medida que se tienen más atributos diferentes de nulo en la Base de Datos. Consecuentemente puede ser útil realizar estos tipos de consultas más de una vez.

Se hace necesario una vez explicado el mecanismo general destacar que dado que las consultas realizadas tienden a ser por su naturaleza de menor complejidad que las consultas originales, es posible realizarlas como una consulta más en ODIL. Esto es, si se tiene por ejemplo una consulta con agregados (caso que se verá posteriormente), el mecanismo de determinación de la incompletitud trabajará sobre los miembros del mismo, por lo que la composición de la respuesta que realiza el procesador de consultas que hace ODIL no afecta de forma alguna la determinación del origen de incompletitud.

2.2. Forma canónica de los diferentes tipos de consulta.

Como se ha visto, el proceso de determinar la incompletitud de una respuesta por la falta de valores de atributos, incluye la realización de consultas adicionales sobre los conceptos y atributos que participan en la consulta hecha para precisar la incompletitud. A continuación se presentan varios casos diferentes para poder identificar fácilmente dada la forma general de una consulta que consulta se debe realizar.

2.2.1. Consulta simple

La forma más simple la siguiente:

LIST <Lista de atributos L1>INSTANCES
SUCH THAT <Calificación de la consulta con condiciones sobre la lista de atributos L2>

Que genera dos tipos de consultas adicionales diferentes para cada uno de los conceptos involucrados, para determinar incompletitud, a saber:

1) Para cada atributo A perteneciente a L1:

LIST <Clave del Concepto>INSTANCES
SUCH THAT <A is null, Calificación de la consulta con las condiciones sobre la lista de atributos L2>;

2) Para cada atributo A perteneciente a L2:

LIST <Clave del Concepto>INSTANCES
SUCH THAT <A is null, Calificación parcial de la consulta con las condiciones sobre L2-Atributo A>;

Para ilustrar este caso, considérese la siguiente consulta en ODIL:

LIST pais.poblacion INSTANCES
SUCH THAT pais.capital = 'Brasilia';

Si la tupla correspondiente a Brasil no tiene un valor no nulo para el atributo capital la siguiente consulta dará una lista vacía, considerando que en la Base de Datos se tiene lo siguiente:

Nombre	Capital	Población
Brasil	Null	5.000.000

La consulta pertinente para determinar la incompletitud en este problema sería:

LIST pais.nombre INSTANCES
SUCH THAT pais.capital is null;

y

LIST pais.nombre INSTANCES
SUCH THAT pais.poblacion is null AND pais.capital = 'Brasilia';

La primera consulta permite determinar que la capital de Brasil tiene un valor nulo. Análogamente, en caso de que la población tuviese un valor nulo, la segunda consulta determinaría que no se tiene la población de Brasil.

2.2.2. Consulta con funciones agregadas

Son del tipo:

LIST <Función agregada sobre atributos L0, Lista de atributos L1>INSTANCES
SUCH THAT <Calificación de la consulta con las condiciones sobre la lista de atributos L2>

Que genera las siguientes consultas adicionales para determinar incompletitud (para cada concepto que participe):

1) Para cada atributo A de L0

LIST <Clave del Concepto>INSTANCES
SUCH THAT <A is null Calificación de la consulta con las condiciones sobre la lista de atributos L2>;

2) Para cada atributo A de L1:

LIST <Clave del Concepto>INSTANCES
SUCH THAT <A is null, Calificación de la consulta con las condiciones sobre la lista de atributos L2>;

3) Para cada atributo A perteneciente a L2:

LIST <Clave del Concepto>INSTANCES
SUCH THAT <A is null, Calificación parcial de la consulta con condiciones sobre L2-Atributo A>;

En esencia las consultas generadas son muy similares a las consultas generadas en el caso simple, sólo que se añade la verificación de nulos sobre los agregados.

Para consultas con funciones agregadas, se puede presentar este ejemplo:

LIST sum (pais.poblacion) INSTANCES
SUCH THAT pais.nombre like 'A' OR pais.nombre like 'B' ;

Genera las consultas:

LIST pais. nombre_pais INSTANCES
SUCH THAT pais.poblacion is null AND (pais.abreviacion like 'A %' OR pais.nombre abreviacion like 'B %') ;

La cual permite estar seguro de tener todas las poblaciones necesarias para calcular el agregado. También se genera la consulta:

LIST nombre_pais INSTANCES
SUCH THAT pais.abreviacion is null;

Que permite ver que todas las abreviaciones estén completas.

Si se tienen los siguientes datos en la BD:

País	Población	Abreviación
Venezuela	30	null
Argentina	null	Arg
Colombia	40	null

La primera consulta determina que falta la población de Argentina. La segunda determina que falta la abreviación de Venezuela.

2.2.3. Consultas anidadas.

Son del tipo:

LIST <Lista de atributos L1>INSTANCES
SUCH THAT <Condiciones sobre la lista de atributos L2, Subconsulta LIST <Lista de atributos L3>INSTANCES
SUCH THAT <Condiciones sobre la lista de atributos L4>>

Que genera las siguientes consultas adicionales para determinar incompletitud:

1) Para cada atributo A de L1

LIST <Clave del Concepto>INSTANCES
SUCH THAT <A is null AND Calificación con las condiciones de consulta anidada y la lista de atributos L2>;

2) Para cada atributo A perteneciente a L2:

LIST <Clave del Concepto, Atributo A>INSTANCES
SUCH THAT <A is null AND Calificación parcial de la consulta con las condiciones de la consulta anidada-Atributo A U L2-Atributo A>;

3) Para cada atributo A de L3

LIST <Clave del Concepto>INSTANCES
SUCH THAT <A is null AND Calificación de la consulta con las condiciones sobre lista de atributos L4>;

4) Para cada atributo A perteneciente a L4:

LIST <Clave del Concepto>INSTANCES
SUCH THAT <A is null AND Calificación parcial de la consulta con las condiciones sobre la lista L4-Atributo A>;

El orden de resolución de estas consultas es similar al orden de resolución de consultas del caso más simple y general. En efecto, puede verse como un proceso Bottom-Up en el cual el árbol de la consulta se va resolviendo de abajo hacia arriba.

Por ejemplo, en la siguiente consulta se pide el clima de las regiones cuyas poblaciones totales son mayores a 100.000 habitantes:

LIST region.clima INSTANCES
SUCH THAT 100000 <(list region^pais.poblacion_pais instances)
AND (region.abreviacion like 'A' or region.abreviacion like 'L*') AND region.continente =*
'AMERICA';

En este caso hay que realizar consultas para determinar que atributos faltan tanto en la consulta interior como en la consulta exterior:

1) LIST region.nombre INSTANCES
SUCH THAT region.clima is null;

2) LIST region.nombre INSTANCES
SUCH THAT region.continente is null AND 100000 <(list region^pais.poblacion_pais instances)
AND (region.abreviacion like 'A' or region.abreviacion like 'L*)*

LIST region.nombre INSTANCES SUCH THAT
region.abreviacion is null AND 100000 <(list region^pais.poblacion_pais instances) AND
region.continente = 'AMERICA';

3) LIST nombre_pais, INSTANCES SUCH THAT pais.poblacion is null;

Si se tienen los siguientes datos en la Base da Datos:

Nombre de región	Clima	Continente	Abreviacion
Andina	Tropical	America	And
Latinoamerica	null	America	null

Nombre	Poblacion
Venezuela	null
Brasil	50
Colombia	35

Y se tiene que la región andina está formada por Venezuela, Colombia y la región latinoamericana por Venezuela y Brasil. Entonces las consultas devuelven los siguientes resultados:

1) Latinoamerica

2) vacia.

Andina, And.
Latinoamerica, null. (se determina que no se tiene la abreviación de latinoamerica).

3) Venezuela

Así, las tres consultas realizada permiten determinar los atributos que están incompletos: la población de Venezuela (necesaria para el agregado) y el clima y abreviación de la región Lationamérica.

2.2.4. Consultas con agregados

El uso de agregados de la ontología puede ser manejado de forma similar a la manera como se aborda el uso de funciones agregadas. Es necesario verificar en primer lugar que las partes del agregado que se define esté completo, esto se puede hacer con las siguientes consultas:

LIST <Lista de atributos L0 de conceptos que son partes del agregado, Lista de atributos L1>INSTANCES
SUCH THAT <Calificación de la consulta con las condiciones sobre la lista de atributos L2>

Que genera las siguientes consultas adicionales para determinar incompletitud:
1) Para cada atributo A de L0 (lista de atributos del concepto parte)

LIST <Clave del Concepto parte del agregado>INSTANCES
SUCH THAT <A is null AND Calificación de la consulta con condiciones sobre la lista de atributos L2>;

2) Para cada atributo A de L1

LIST <Clave del Concepto sobre el que se hace la consulta>INSTANCES
SUCH THAT <A is null AND Calificación de la consulta con condiciones sobre la lista de atributos L2>;

3) Para cada atributo A perteneciente a L2:

LIST <Clave del Concepto sobre el que se hace la consulta>INSTANCES
SUCH THAT <A is null AND Calificación parcial de la consulta con condiciones sobre L2-Atributo A>;

Aunado a esto, existe otro origen de incompletitud para este subcaso que se debe contemplar. Puede darse el caso de que el agregado no esté completo. Es decir, puede ser que falten instancias en la lista de los miembros del agregado. Por ejemplo si se tiene un concepto "Equipo" que es un agregado de "Jugadores", puede ser que falten jugadores en un equipo.

Desafortunadamente este problema no puede ser resuelto con el mecanismo propuesto para resolver incompletitud por la falta de valores de atributos. En cierta forma, es comparable con el problema de

descubrir interrelaciones entre conceptos a partir de la extracción, ya que el pertenecer a un agregado es una interrelación.

Por ejemplo se tiene el siguiente caso:

LIST region^pais.poblacion_pais, region.clima INSTANCES
SUCH THAT region.abreviacion like 'And';

Se generan las siguientes consultas:

1) LIST pais.nombre INSTANCES
SUCH THAT pais.poblacion_pais is null;

2) LIST region.nombre INSTANCES
SUCH THAT region.clima is null AND region.abreviacion like 'And';

3) LIST region.nombre INSTANCES
SUCH THAT region.abreviacion is null;

Si se tienen los siguientes datos:

Nombre	Población	Región
Venezuela	35	Andina
Colombia	null	Andina
México	20	Norteamerica

Nombre	Abreviacion	Clima
Andina	And	Tropical
Nortamerica	null	null

Las consultas devuelven:

1) Colombia (falta su población).
2) Nortamerica. (falta el clima)
3) Norteamerica. (falta la abreviacion).

2.2.5. Consultas con relaciones de especialización-generalización.

Como se ha estudiado anteriormente, si se parte de la suposición de que al realizar una consulta sobre una superclase o una subclase las consultas realizadas sobre la base de datos tomen consideración la estructura jerárquica de la generalización-especialización entonces no se hace necesario tomar ningún tipo medidas adicionales para las relaciones de especialización-generalización. Esto es, si el procesador de consultas ODIL enmascara la relación, no es necesario realizar ninguna operación extra. Es importante no obstante tomar en cuenta que la clave de una subclase puede incluir atributos de la superclase.

Capítulo 3

Diseño

La Extracción Focalizada de Información supone asimismo tres fases similares a las que plantea DIA. En primera instancia es necesario determinar el conjunto de documentos sobre el cual se realizará la FIE. El conjunto de documentos puede ser heterogéneo. Es decir, puede ocurrir que la información necesaria para encontrar información faltante sobre un concepto se encuentre en más de un tipo de documento. Puede pasar de igual manera que un tipo documento sirva para responder consultas sobre más de un concepto de la ontología y sobre más de una instancia. Puede ocurrir de igual manera que los documentos contengan información adicional que no está relacionada con los dominios de información de los conceptos de la ontología. En tal sentido, la primera labor necesaria para la FIE consiste en definir el conjunto de documentos y cómo acceder a ellos.

Una vez definido el conjunto de documentos, es necesario construir extractores de información que permitan encontrar la información faltante en la ontología. Igual que en el esquema general de DIA, el extractor de información puede realizarse utilizando mecanismos de procesamiento de lenguaje natural, de aprendizaje de máquina, etcétera. Lo relevante en esta fase de DIA es que el extractor de información debe construirse tomando en cuenta la información contenida en la ontología, el *contexto de extracción*. Dicho contexto se puede utilizar para facilitar la búsqueda de la información faltante.

En tercer lugar es necesario determinar los orígenes de incompletitud. Esto se debe realizar tomando en consideración dos cosas: la consulta realizada por el usuario y la información ya existente en la ontología. La incompletitud puede abarcar más que valores necesarios para responder las consultas: puede incluir información necesaria para calificar la consulta. Tomando en cuenta estos dos elementos se pueden realizar tantas búsquedas de información como sean necesarias.

La determinación de los origenes de incompletitud de información es un tema que será abordado posteriormente en profundidad. Sin embargo es importante precisar en este punto que pueden existir más de

un tipo de fuentes de incompletitud:

1. Incompletitud por la ausencia de instancias (tuplas) en la ontología.
2. Incompletitud por el desconocimiento de valores de atributos.
3. Incompletitud por el desconocimiento de interrelaciones entre instancias.

Para este trabajo se ha definido el alcance de la Extracción Focalizada de Información en la determinación y resolución de incompletitud por desconocimiento de valores de atributos. El motivo de esto es que el descubrimiento de nuevas tuplas es un problema que no puede ser resuelto por Extracción Focalizada de Información, ya que la misma supone la existencia de información en la ontología sobre todas las instancias involucradas en una consulta. Por otro lado, si bien el descubrimiento de interrelaciones entre instancias puede modelarse como el descubrimiento de los valores de las claves foráneas que definen cada interrelación, se ha determinado que esta tarea es más compleja y puede involucrar tareas de extracción más sofisticadas que salen del alcance de este proyecto.

Se supone, por lo tanto, que todas las instancias concernientes a los dominios sobre los cuales se hace la búsqueda están contenidas en la ontología con un subconjunto de sus campos con valores conocidos y que las interrelaciones entre los conceptos presentes en la ontología están determinadas.

Finalmente, para probar la solución propuesta para la FIE, se trabajó con tres dominios de información relacionados con el funcionamiento interno de la Universidad Simón Bolívar: las designaciones de cargos, los ingresos y ascensos dentro del escalafón de Profesores y la designación de Jurados para Trabajos de Ascenso. Para trabajar con esos 3 dominios se trabajó con las Actas de Consejos Directivos y Actas de Consejos Académicos como conjuntos de documentos. La información concerniente a las designaciones de cargos, los ingresos y ascensos dentro del escalafón de Profesores se puede encontrar en las Actas de los Consejos Directivos. Asimismo, la información concerniente a la designación de Jurados para Trabajos de Ascenso se encuentra en las Actas de Consejos Académicos.

La tarea de Extracción Focalizada de Información está compuesta por dos grandes labores: la Determinación del Origen de Incompletitud (DOI) y la Búsqueda Focalizada de Información (BFI). La Determinación del Origen de Incompletitud busca, como se ha dicho anteriormente, encontrar qué información está incompleta dentro de la ontología y en la base de datos relacional. Los objetivos de la misma deben ser seleccionar un dato que se quiera obtener en la búsqueda focalizada y construir un contexto que facilite la búsqueda. La Búsqueda Focalizada de Información toma los resultados de la Determinación del Origen de Incompletitud y busca la información faltante en un conjunto de documentos relacionados al dominio de información de los conceptos involucrados.

A continuación se describirá en detalle esta solución propuesta para realizar ambos pasos. Ambas tareas son hechas por dos módulos diferentes: el Determinador del Origen de Incompletitud y el Buscador Focalizado de Información. Se hablará de cada uno de ellos y se harán algunos comentarios sobre el contexto de extracción que es el producto intermedio entre ambos módulos.

3.1. El Determinador del Origen de Incompletitud (DOI)

El Determinador del Origen de Incompletitud es el módulo que se encarga de determinar qué falta en la ontología para poder responder a una consulta hecha por el usuario. Ya se ha explicado anteriormente el mecanismo general para determinar origenes de incompletitud de atributos, en el capítulo 2. Sin embargo, hay algunas consideraciones generales que son pertinentes mencionar.

Como se ha visto, el mecanismo general para determinar los origenes de incompletitud por el desconocimiento de valores de atributos consiste en realizar varias consultas que involucran la clave primaria de los conceptos involucrados. Las consultas tienen como objetivo encontrar valores nulos que indican incompletitud. Ahora bien, para poder hacer esto, es necesario realizar algunas transformaciones de la consulta. En primera instancia se hace necesario transformar las condiciones de la consulta a la Forma Normal Conjuntiva. Esto es importante para poder eliminar restricciones sobre los atributos sobre los cuales se quiere determinar si hay incompletitud. En segunda instancia es necesario también aplanar la consulta resolviendo consultas anidadas y las operaciones presentes en las condiciones. Este paso es necesario para poder construir los contextos de extracción. En tercer lugar es necesario ejecutar cada una de las consultas según lo especificado en el capítulo 2.

Algo que es conveniente destacar es que el acceso a la Base de Datos tuvo que ser simulado. Esto se debe a que el software existente para procesar consultas en ODIL, creado por Mirisola y Apaza [4] fue desarrollado utilizando librerías que ya no están mantenidas y que fueron imposibles de utilizar. Como el alcance de este proyecto de grado no incluye el procesamiento de consultas de ODIL, se simuló el acceso a la ontología - esto es, la interpretación, ejecución y respuesta de las consultas - sin utilizar una base de datos ni consultas escritas en el lenguaje ODIL.

Duda: ¿Es pertinente en este capitulo el ultimo parrafo?

3.1.1. El Contexto de Extracción

Como se ha dicho, la tarea del Determinador del Origen de Incompletitud es determinar qué información falta para responder una consulta y preparar un contexto de extracción que facilite la búsqueda focalizada de información. El contenido del contexto de extracción puede incluir valores de atributos que estén relacionados con la información que falta, puede incluir posibles valores de atributos que falten, sinónimos del mismo, heurísticas sobre cómo buscar el valor en el conjunto de documentos y en general cualquier información que pueda ser útil para realizar la búsqueda focalizada. Lo importante en todo caso es que el contexto de extracción sea preparado tomando en cuenta el tipo de extractor que se utilice para la búsqueda focalizada, el tipo de documento, y el dominio de información sobre el cual se realice la extracción.

En el caso particular de este proyecto, el contexto de extracción contiene los valores de atributos que están relacionados con el atributo cuyo valor se quiere ubicar. Esto se hizo tomando en cuenta varios aspectos sobre los dominios sobre los cuales se hace la extracción. En primer lugar, los tres dominios de información sobre los cuales se realiza la extracción son relativamente sencillos y poco complejos en estructura. Además de ésto, los datos pueden agruparse en *unidades de información*, fragmentos de texto que contienen toda la información relacionada con todos los conceptos del dominio.

Por ejemplo, un posible dominio de información es la designación de un jurado de ascenso dentro de la Universidad Simón Bolívar. La información concerniente a este dominio puede ser encontrada en las actas de los consejos directivos. En este contexto, la unidad de información vendría siendo los párrafos, tablas, enumeraciones, entre otros, que se refieren a la designación de un jurado de ascenso. Dichas unidades son relativamente fáciles de identificar en los documentos y por lo tanto la tarea de la extracción consiste en determinar cuál es la unidad de información relacionada con el atributo que falta para después ubicar en cada unidad el valor faltante. En tal sentido, la identificación de dicha unidad puede realizarse apoyándose en los valores de atributos conocidos.

De manera que para este proyecto se concibe un contexto de extracción como un conjunto de valores de campos relacionados con el atributo cuyo valor falta. Es relevante también describir cómo se obtienen esos valores faltantes, ya que existen varias formas de hacerlo. Por ejemplo, si se conoce el valor de la clave primaria o de un conjunto de atributos que permita identificar un conjunto relativamente pequeño de instancias, podrían hacerse consultas adicionales para obtener los valores del resto de los atributos. Esto supone por supuesto realizar consultas adicionales a la base de datos y puede no ser útil cuando el conjunto de instancias que se puede seleccionar con los atributos conocidos es muy grande.

Alternativamente, podrían pasarse en lugar de los valores de los atributos que faltan las condiciones especificadas en la consulta hecha por el usuario y que en lugar de ubicar cadenas de caracteres ya cono-

cidas en los documentos se extraigan los valores del documento y se verifiquen las condiciones. La ventaja de este último mecanismo es que no se hace necesario determinar un conjunto de posibles valores y el contexto de extracción es más reducido en tamaño. La desventaja es que se depende de tener un buen extractor que identifique en cada unidad de información el valor de todos los atributos existentes: si el valor está en la unidad pero no se identifica se puede pasar por alto.

En todo caso, lo importante a destacar es que existe más de una forma posible de construir un contexto de extracción que se base en valores de atributos conocidos. Para este proyecto, el contexto fue construido elaborando una lista de valores posibles a partir de las condiciones de la consulta hecha por el usuario. En los casos en los que se tienen restricciones de igualdad a uno o más valores (condiciones = o IN) es fácil determinar el posible valor del campo. En los casos en los que se tienen restricciones que operan sobre un rango, el problema es relativamente más complejo ya que es necesario determinar un *conjunto* de posibles valores del campo. Para ello se decidió especificar como parte de la configuración del sistema conjuntos de valores que pueden tener en general cada uno de los atributos y se realiza una intersección con las condiciones de la consulta para determinar, para cada consulta (y consecuentemente cada contexto de extracción), los posibles valores. Dichos conjuntos pueden ser definidos explícitamente (listando los valores posibles) o implícitamente (definiendo un intervalo de valores posibles y alguna forma de generar todos los valores posibles por medio, por ejmplo, de un incremento).

La principal desventaja de este mecanismo es que su funcionamiento depende de que se puedan generar correctamente todos los posibles valores de cada uno de los atributos. Si falta alguno de los valores posibles, es factible que no se identifique la unidad de información apropiada. La ventaja y en esencia el motivo por el cual se prefirió este mecanismo sobre los ya listados anteriormente, es que no se depende del extractor focalizado ni de la información que se conozca; siempre se generan todos los posibles valores si la configuración del sistema es apropiada.

Por último, algo que debe tomarse en cuenta es que a menudo las restricciones dadas por las condiciones en una consulta pueden incluir ORs en los que se especifican condiciones relativamente complejas. El contexto de extracción debe ser construido de forma que permita reflejar todos los posibles estados que satisfagan las condiciones. Dicho espacio puede ser visto como el producto cartesiano de los conjuntos determinados por los átomos presentes en las cadenas de OR de la Forma Normal Conjuntiva. Esto es, entendiendo como *átomo* una condición que espeficique una restricción puntual sobre uno o más atributos (igualdad, mayor o menor, etcétera), cada *elemento* de la condición en la NCF puede ser una cadena de ORs que operen sobre varios átomos o por el contrario un átomo individual. Un contexto que consista en los posibles valores de los atributos debería contemplar todo el espacio generado. Por ejemplo, si la condición es de esta forma:

$$(C_{1,1} \vee C_{1,2} ... \vee C_{1,N}) \wedge (C_{2,1} \vee ... C_{2,N}) \wedge (C_{M,1} \vee ... C_{M,N})$$

Donde $C_{m,n}$ se refiere al átomo número n de N del elemento número m de M en la CNF.

Los posibles estados que pueden generarse son de la forma

$$(C_{1,1}, C_{2,1}, ..., C_{M,1}), (C_{1,2}, C_{2,1}, ..., C_{M,1}), ...(C_{1,N}, C_{2,N}, C_{M,N})$$

Cada estado representa un posible conjunto de valores que satisfacen las condiciones especificadas en la consulta. En tal sentido, cada uno de estos estados debería ser evaluado por separado al determinar cuál es la unidad de información correcta.

3.2. El Buscador Focalizado de Información (BFI)

El Buscador Focalizado de Información es el encargado de encontrar los valores de los campos cuyos valores son desconocidos para una consulta dada. Dichos valores son buscados en un conjunto de documentos. En el caso de este proyecto de grado, se requiere que los documentos estén presentes en el sistema de archivos donde se ejecute el sistema utilizando información de contexto proporcionada por el Determinador del Origen de Incompletitud (DOI). Sin embargo, como se ha dicho, la arquitectura DIA no presenta restricciones sobre la naturaleza de los documentos y podrían ser accedidos remotamente e inclusive utilizarse la World Wide Web.

En el caso del presente proyecto de grado, para llevar a cabo esta tarea, el BFI tiene dos componentes: un preprocesador de texto y un extractor focalizado de información.

El Preprocesador de Texto tiene como objetivo preparar el conjunto de documentos sobre el cual se hará la Búsqueda Focalizada de Información y dejarlo listo para la realización de la extracción. Las tareas para ello incluyen leer el documento en su formato original y extraer el fragmento o el conjunto de fragmentos del documento que están relacionados con el dominio sobre el cual se hará la Búsqueda Focalizada de Información.

En este proyecto de grado, los dominios elegidos se caracterizan por tener un conjunto de documentos bastante homogéneos y el preprocesamiento consiste en esencia en elegir los fragmentos de documentos que están relacionados con el dominio informativo sobre el cual se hará la extracción focalizada. Sin embargo, en general el preprocesamiento de documentos podría incluir tareas más amplias: puede darse que el conjunto de documentos sea heterógeneo y que se requiera una preparación adicional; puede darse

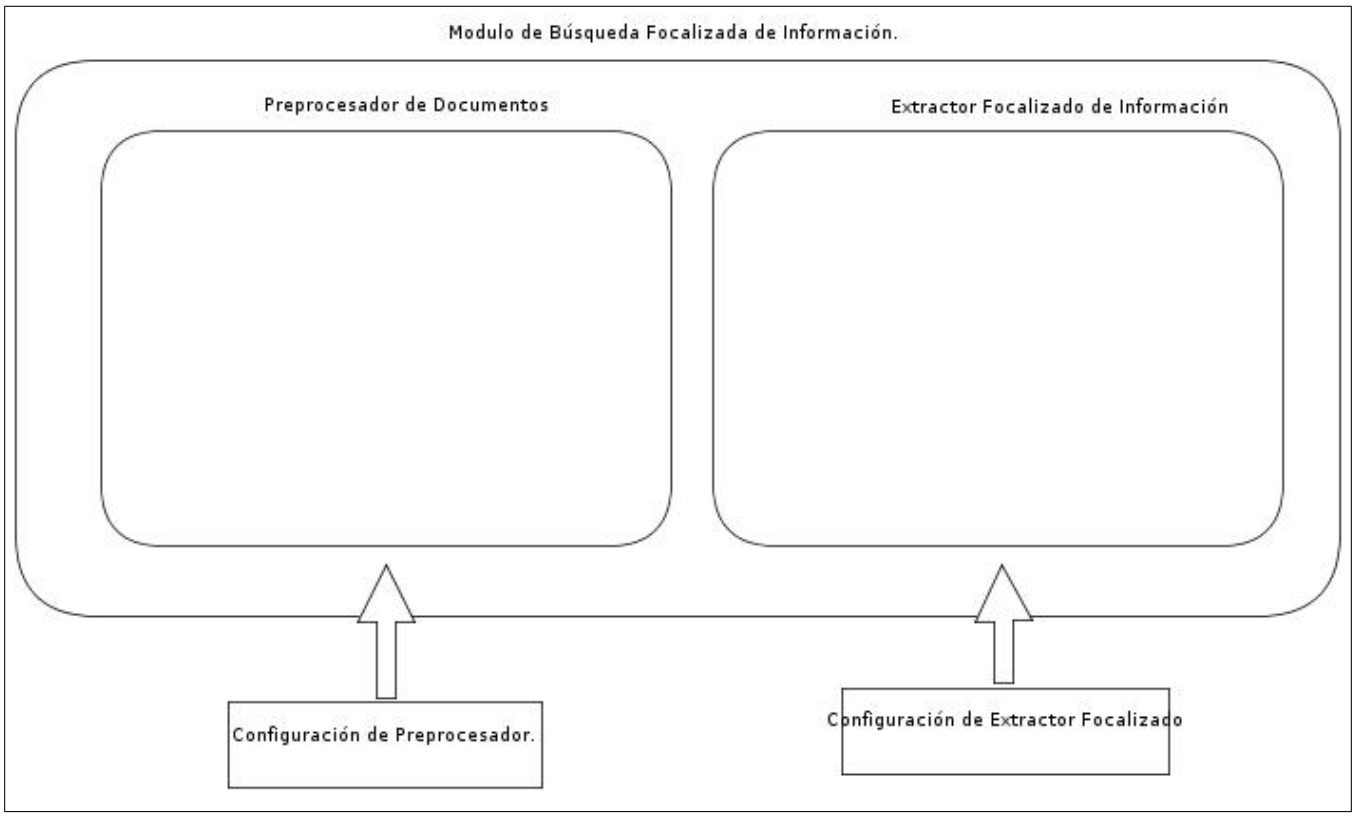


Figura 3.1: Diagrama del Buscador Focalizado de Información

analógamente que un mismo documento tenga información de diferentes dominios que se quieran trabajar. Puede ocurrir también que los documentos esten en más de un idioma. En todo caso lo relevante en todo esto es que el preprocesamiento es una labor que tiene como objetivo preparar a los documentos para realizar la extracción propiamente dicha y que puede incluir muchas subtareas.

De igual forma, es importante destacar que el preprocesamiento de texto es concebido como una tarea que se realiza *offline*. Es decir, cada conjunto de documentos puede ser preprocesado previamente de forma para que las consultas sean respondidas lo más rápidamente posible.

Por su parte, el Extractor Focalizado de Información (EFI) tiene como objetivo extraer el valor del campo faltante utilizando como materia prima el conjunto de documentos preprocesados y un contexto de extracción. El contexto de extracción, como se ha dicho anteriormente, consiste en toda la información que el DOI puede recopilar sobre la consulta que se está realizando que puede ser de útil para hacer la extracción focalizada de información.

La tarea de extracción es probablemente la que tiene una mayor complejidad asociada y depende

muchísimo del tipo de documento que se trate y de que tan estructurado es. Cómo se ha dicho, en la medida en la que los textos sean mas estructurados - esto es, en la medida en la que el estilo sea regular y homogéneo - se hace más fácil la construcción del extractor. Sin embargo, esto supone un reto abierto en el mundo de la computación y los avances más recientes en el área de IE incluyen extractores basados en modelos estocásticos y Machine Learning.

Anteriormente se han diseñado extractores para DIA con resultados bastante buenos. Ruiz (*año*) [2] construyó un extractor basado en Hidden Markov Models (Modelos Ocultos de Markov) con resultados muy buenos. En el caso particular de este proyecto, no obstante, el objetivo principal no es la construcción de un extractor focalizado sofisticado. El extractor focalizado concebido trabaja con expresiones regulares, que buscan explotar al máximo la estructura de los documentos. En esencia, lo que se tiene para ello son una serie de reglas de extracción que especifican delimitadores anteriores y posteriores a la unidad de texto que se quiere extraer. Dicha tecnología se utiliza en este proyecto para el preprocesador de texto y para el extractor focalizado.

El motivo por el cual se eligió el uso de expresiones regulares es que los dominios de información tienen documentos semiestructurados, que en general tienen un estilo de diagramación y redacción que permanece bastante similar. Esto es, es posible identificar frases y palabras que pueden servir de delimitadores para encontrar los datos requeridos. Para ello, es necesario que un usuario experto pueda identificarlas y compilar una lista de expresiones regulares. Esta labor no puede ser automatizada y debe ser realizada manualmente para cada dominio de información y atributo. En todo caso, es importante destacar que cualquier extractor focalizado requiere intervención humana para poder configurar la manera como se extrae la información, bien sea porque involucra marcar documentos (Document Tagging) o porque requiera aprendizaje de máquina en el cual se requiere una configuración mínima por parte del usuario.

Es importante aclarar que la solución propuesta no depende del tipo de extractor que se construya. En general DIA y la FIE definen pasos necesarios para interrogar documentos; la construcción del extractor es una tarea que depende fundamentalmente del dominio sobre el cual se trabaje.

A manera de conclusión, es relevante decir que el DOI trabaja directamente con el Extractor Focalizado de Información para encontrar los valores faltantes. Esto es, una vez que se determinan los origenes de incompletitud, el EFI recibe solicitudes de búsqueda para cada uno de los campos con valores faltantes. Esas solicitudes permiten descubrir y completar la ontología y dar respuesta a las consultas que tiene el usuario. El preprocesador se ejecuta una única vez por conjunto de documentos.

Capítulo 4

Detalles de implementación

4.1. Consideraciones generales

4.1.1. Lenguaje de programación

El lenguaje de programación utilizado para la implementación del sistema de Extracción Focalizada de Información fue Java en su versión 1.6.0_20.

4.1.2. Librerías utilizadas

Se utilizaron las siguientes librerías open source:

- Para desplegar mensajes de traza y errores se utilizó la librería Log4J. Dicha librería permite en lugar de imprimir mensajes por el Standard Output, imprimirlos dependiendo de un nivel de desarrollo espeficiado en un archivo de configuración. A la hora de desplegar un mensaje se elige un nivel de desarrolo: traza, error, error fatal, warning, entre otros. Luego con modificar el archivo de configuración de la librería se puede fácilmente apagar y prender los niveles para que se impriman o dejen de imprimir algunos mensajes y cambiar rápidamente entre etapas de desarrollo, debugging, pruebas y puesta en producción. Se utilizó la version 1.2.16.
- Para procesar archivos PDF se utilizó la librería PDFBox y Tika, en sus versiones 1.6 y 1.1.
- Para procesar archivos HTML se utilizó la librería Jericho, en su versión 3.2.
- Para procesar archivos .doc se utilizó la librería POI, en su versión 3.8.
- Para leer archivos de configuración de XML se utilizó la librería que ofrece Java para parsear y consultar XML, JAXP, en su versión 1.4.

4.2. Implementación del Determinador del Origen de Incompletitud (DOI)

El Determinador del Origen de Incompletitud tiene como objetivo encontrar qué valores de atributos desconocidos son necesarios para poder responder a una consulta y construir un contexto de extracción. Las tareas para poder hacer esto incluyen transformar las condiciones a la forma normal conjuntiva, aplanar las consultas, realizar las consultas sobre la ontología y construir el contexto de extracción. A continuación se expondrán algunas consideraciones relevantes sobre estas tareas.

4.2.1. Transformación de las condiciones a la Forma Normal Conjuntiva

Como se ha dicho anteriormente, una condición se encuentra en la Forma Normal Conjuntiva (NCF) [14] cuando consiste en una secuencia conjuntiva (de operadores ANDs) de una o más condiciones de tipo disjuntiva o de átomos -esto es predicados o negaciones de predicados -. Toda afirmación lógica escrita en términos de conjunciones, negaciones y disjunciones puede ser escrita en la forma normal conjuntiva. (Weisstein 1997 [14]).

El procedimiento para transformar cualquier afirmación en la NCF consiste en realizar una serie de conversiones aplicando algunos teoremas lógicos como las Leyes de De Morgan, entre otros. El procedimiento utilizado es una implementación del mecanismo propuesto por el Profesor Jason Eisner. [15].

4.2.2. Implementación del constructor de contextos de extracción

Para poder construir los contextos de extracción, es necesario en primer lugar aplanar las consultas y las condiciones. El motivo de esto es que el contexto de extracción diseñado para este proyecto, tal y como fue especificado en la sección 3.1.1, consiste en los posibles valores conocidos de los atributos relacionados con el atributo cuyo valor falta. Por lo tanto, el primer paso consiste en aplanar las consultas resolviendo cosas como las consultas anidadas, las operaciones aritméticas presentes en las condiciones, operaciones sobre cadenas de caracteres y en general cualquier cosa que sea necesaria para que en cada consulta sólo hayan condiciones sin operaciones adicionales.

Una vez aplanadas las consultas, se realiza la construcción del contexto de extracción. Para ello, se generan todos los posibles valores de cada atributo relacionado con el atributo original siguiendo la especificación obtenida en un archivo de configuración relacionado con cada dominio de información. En dicho archivo de configuración se especifica el dominio explícita o implícitamente, enumerando los posibles valores que puede tener o unas cotas superiores e inferiores y un incremento para generar los posibles

valores entre ellos, útil para dominios numéricos - interos y punto flotante - y fechas. Los posibles valores de cada atributo son generados tomando en cuenta esta especificación y las condiciones de cada consulta.

4.3. Implementación del Buscador Focalizado de Información (BFI)

Como se ha explicado en 3.2, el Buscador Focalizado de Información (BFI) es el módulo que encuentra los valores de los campos cuyos valores son desconocidos o incompletos para una consulta dada. Estos valores son buscados en un conjunto de documentos definidos por el usuario y se encuentran tomando en cuenta un contexto de extracción que contiene información útil para la búsqueda. El BFI tiene dos componentes: un preprocesador de texto y un extractor focalizado de información.

En grandes rasgos, el funcionamiento y la implementación de ambos componentes son bastantes similares. Ambos componentes operan como extractores de texto utilizando expresiones regulares definidas previamente por un usuario experto. Dichas expresiones regulares funcionan como delimitadores del fragmento de texto que interesa según la tarea que se esté realizando: en el caso del preprocesador de texto, el objetivo es extraer el fragmento de texto que está relacionados con el dominio sobre el cuál se realiza la extracción focalizada; el extractor focalizado busca encontrar el valor del campo faltante para responder una consulta.

Ambos componentes están hechos para trabajar con un archivo de configuración en XML que especifica los parámetros necesarios para realizar el preprocesamiento y la extracción. En particular los parámetros que se guardan en estos archivos de configuración incluyen las expresiones regulares, las rutas de directorio de los archivos de entrada, la ruta de los archivos de salida para el caso del preprocesador, entre otras cosas. De esta manera, el usuario del sistema debe preocuparse tan sólo por entender cómo realizar un archivo de configuración para cada uno de los dominios necesarios.

Más adelante se expondrán algunas consideraciones sobre las expresiones regulares de interés para el entendimiento de la implementación del BFI.

4.3.1. Preprocesador de Textos

Ahondando más en el componente de pre-procesamiento, su objetivo como se ha dicho es extraer de cada documento el conjunto de fragmentos que está relacionado con el dominio sobre el cual se hace la búsqueda focalizada. En tal sentido tiene como entrada un conjunto de documentos en su formato original (PDF, Portable Readable Document; HTML , Hyper Text Markup Language; DOC y DOCX, archivos de Microsoft Word) y un archivo de configuración. Su salida es un archivo .txt en el cual se tiene el conjunto

de fragmentos que están relacionados con el dominio en cuestión según lo especificado por las expresiones regulares. Para procesar los documentos se utilizaron las librerías descritas anteriormente.

Empíricamente se observó que el procesador de expresiones regulares toma una cantidad de tiempo considerable cuando no lograr hacer un match. Para poder controlar esto se utilizó un sistema de timeouts basado en la utilización de threads, para agilizar el preprocesamiento y la fase de pruebas.

4.3.2. Extractor Focalizado

En cuanto al componente de extracción focalizada, su objetivo es dado un conjunto de fragmentos de documentos que puede contener valores de campos necesarios para responder una consulta, encontrar el valor de esos campos. Para ello opera en 2 fases: primero rompe cada documento en unidades relacionadas con el dominio en cuestión. Para ilustrar este punto, tomando en cuenta los 3 dominios que se consideraron para este trabajo, las unidades vienen siendo designaciones, ingresos y ascensos dentro del escalafón y designación de un jurado de ascenso, por trabajo. Este refinamiento sobre los documentos es hecho utilizando también expresiones regulares y su objetivo es aislar fragmentos de texto donde haya una cierta cohesión semántica (un fragmento de texto que se refiera a los mismo) que permita ubicar la respuesta utilizando el contexto de extracción.

Una vez separados los documentos en unidades, se utiliza el contexto de extracción para ordenar las unidades en un orden de preferencia. Dicho orden se realiza tomando un valor denominado *UnitHitMeasure* que busca medir a través de una media ponderada cuánto se aproxima o relaciona cada unidad encontrada en los documentos a la unidad correcta según lo especificado por el contexto de extracción. Por ejemplo, si se está trabajando con designaciones de cargos, se buscan ordenar las unidades tomando en cuenta el contexto de extracción y los valores de campos conocidos sobre una designación las designaciones según la presencia de los valores conocidos en cada unidad.

El UnitHitMeasure es calculado de la siguiente manera: cada campo de la ontología tiene un peso que es asignado por un usuario experto en el archivo de configuración. Cada contexto de extracción tendrá uno o más valores de campos conocidos. Para cada campo presente el contexto de extracción cuyo resultado es conocido y que está presente en la unidad se suma el peso asociado a ese campo. Luego se divide esa sumatoria entre la máxima suma posible (el valor de la suma si todos los campos cuyo valor se conocen estuviesen en la unidad con los valores conocidos). De esta manera se obtiene un número del 0 al 1 que indica el grado de relación que tiene la unidad en cuestión con el contexto de extracción focalizado.

Algo que no fue implementado por limitaciones de tiempo es el mecanismo señalado al final de la sección 3.1.1 para generar todos los posibles estados de las condiciones que se derivan de la forma normal

conjuntiva. La forma de implementar de esto consiste en generar una lista de contextos de extracción, cada uno asociado a un posible estado según lo explicado, en lugar de un solo contexto. Luego al hacer la búsqueda focalizada se debe hacer con cada uno de los contextos de extracción y usar un mecanismo para discriminar cuál es la respuesta correcta.

DUDA: Profe, quiere que coloque un ejemplo de cómo se calcula esto para que quede más claro? O se entiende para un lector cualquiera?

Vale acotar que a la hora de determinar si un valor está presente en la unidad se pueden realizar modificaciones del String con el objetivo de incorporar el uso de sinónimos, traducción de idiomas, desglose de siglas, entre otros, que pueden permitir aumentar la posibilidad de que un campo tenga esté contenido dentro de una unidad de una forma equivalente en cuando a significado. Esto debe ser sin embargo implementado según el contexto y requiere una modificación en los archivos XML, en su lector y la implementación del código para realizar las modificaciones. Por ejemplo, se implementó una funcionalidad para dada una fecha en un formato cualquiera, generar las formas de fechas más comunes según lo utilizado en los documentos.

Una vez ordenadas las unidades en orden de precedencia se ubica el valor del campo que se quiere encontrar por medio de expresiones regulares. Cabe destacar que el usuario puede especificar un *Minimum HitMeasure* (valor mínimo de HitMeasure) para que una respuesta sea considerada válida: mientras más cercana a 1 será más probable que la respuesta sea correcta aunque se disminuye la posibilidad de encontrar el valor.

En la figura 4.3.2 puede apreciarse el funcionamiento interno del BFI.

4.3.3. Diseño de las expresiones regulares para la configuración del Preprocesador de textos y del Extractor Focalizado

La elaboración de las expresiones regulares es una tarea que debe realizar un usuario experto que conozca el dominio sobre el cual se quiere realizar la extracción focalizada. En general los dominios admisibles son aquellos con textos estructurados o semi-estructurados, donde se puedan deducir patrones en la redacción de los contenidos. Aún teniendo cierta regularidad en la forma del texto, existen retos como errores de redacción, ortografía, transcripción, formato en los documentos fuente, presencia de tablas entre otros, que dificultan el uso de expresiones regulares. Es posible también por ejemplo que hayan presentes un conjunto de carácteres que parezcan espacios en blanco pero que no hagan match con una expresión regular porque no son espacios.

En todo caso, lo relevante en esta descripción de la implementación del sistema es que para atender esta dificultad los archivos de XML permiten contener varias expresiones regulares ordenadas por orden de precedencia. Los textos se exploran con las expresiones regulares según el orden de precedencia: si no se encuentra un fragmento de texto con una expresión regular se explora posteriormente con otra de menor precedencia y así sucesivamente.

El objetivo de esta precedencia es que el usuario pueda colocar con alta precedencia expresiones regulares que sean bastante selectivas y restrictivas, con el objetivo de asegurar correctitud. Luego a medida que se desciende en la escala de precedencia se puede relajar la restrictividad de la expresión regular, con el objetivo de aumentar la probabilidad de encontrar un fragmento de texto. Relajar la restrictividad implica que la respuesta no sea exactamente la buscada y que por el contrario pueda contener caracteres de más (esto es, que el fragmento de texto obtenido contenga al fragmento de texto correcto). Esto puede ser deseable en muchos casos, dada la complejidad de utilizar expresiones regulares.

DUDA: PONER COMO SE OBTIENEN LAS REGEXPS?

En el Apéndice A se muestran fragmentos de archivos de configuracion.

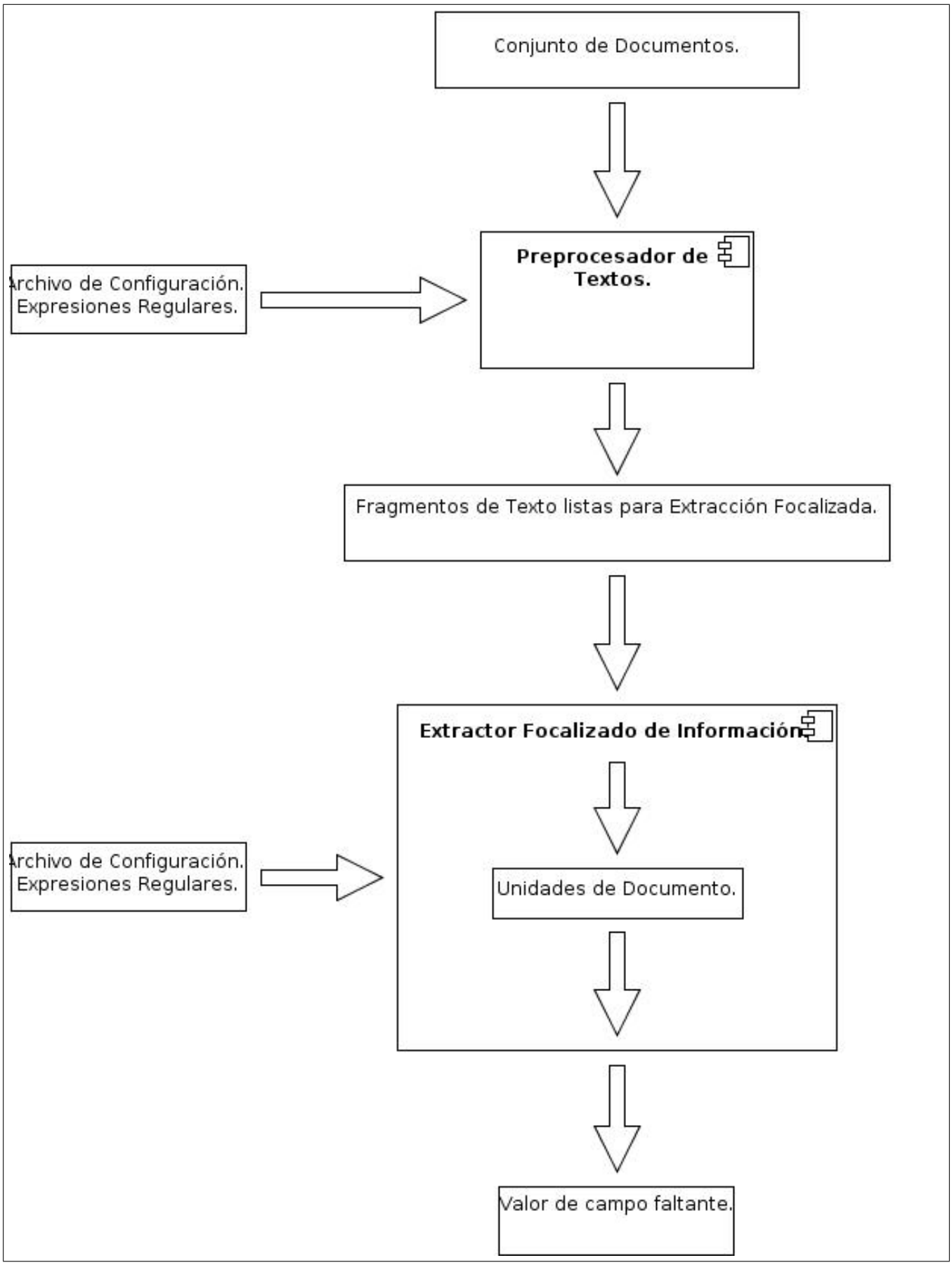


Figura 4.1: Funcionamiento interno del Buscador Focalizado de Información

Capítulo 5

Pruebas

5.1. Consideraciones Generales

Con el objetivo de realizar las pruebas sobre el sistema se trabajó como ya se ha dicho sobre tres dominios relacionados con el funcionamiento interno de la Universidad Simón Bolívar: las designaciones de cargos, los ingresos y ascensos dentro del escalafón de Profesores y la designación de Jurados para Trabajos de Ascenso. Para trabajar con esos 3 dominios se trabajó con las Actas de Consejos Directivos y Actas de Consejos Académicos. La elección de los 3 dominios se tomó buscando tener una variedad de estilos de texto semiestructurado para obtener resultados diversos sobre el funcionamiento.

La selección de los documentos se realizó sobre el conjunto de las actas de los Consejos Directivos y Académicos de la Universidad Simón Bolívar desde el año 1998 hasta el año 2012. Esto permite que las pruebas se hagan sobre un conjunto de documentos que pueden tener variabilidad en su estilo, redacción y estructura en los años. Del conjunto de actas en el intervalo de tiempo especificado, se hizo una preselección de los documentos para utilizar las actas de los Consejos Ordinarios. Adicionalmente, según cada dominio de aplicación se extrajeron las actas que sí contienen información sobre los dominios. Es decir, se utilizaron los documentos que contienen información sobre los dominios elegidos, descartando aquellas actas que no contenían texto de interés.

Una vez definidos los dominios se definieron varios conjuntos de prueba. En primera instancia se probaron el Determinador de Origen de Incompletitud y el uscador Focalizado de Información por separado. Luego se hicieron pruebas sobre ambos compontentes integrados. A continuación se describen con mayor profundidad las pruebas realizadas sobre el sistema.

5.2. Pruebas unitarias sobre el Determinador de Origen de Incompletitud.

5.3. Pruebas unitarias sobre el Buscador Focalizado de Información

Para probar el Buscador Focalizado de Información se hicieron diseñaron 2 pruebas por dominio. Por un lado de hicieron pruebas para el Preprocesador de Texto y en segunda instancia Pruebas para el Extractor Focalizado.

5.3.1. Pruebas del Preprocesador de Texto

Para probar el preprocesador, se tomó un conjunto de archivos seleccionados aleatoriamente con uniformidad sobre los años. La muestra elegida fue de un tercio de la población total de actas existentes, para cada dominio. Para las designaciones se trabajó con 73 documentos, para las incorporaciones y ascensos dentro del escalafón se trabajó con 77 documentos y para la designación de Trabajos de Jurado de Ascenso se trabajó con 49 documentos.

Las pruebas en esencia consistieron en ver si el fragmento de texto extraído por el preprocesador coincidía exactamente, estaba contenido, contenía o era completamente disjunto con el segmento de texto que debería ser extraído. Esto es: pueden verse ambos fragmentos de textos como secuencias de caracteres y se puede examinar si ambas secuencias coinciden por completo, tienen relaciones de contención o son completamente diferentes.

Esta prueba se hizo manualmente: se examinó el resultado de la extracción realizada por el preprocesador y se buscó dentro de cada documento el fragmento que debería extraerse.

Se definió como *aprobado* que o bien el fragmento extraído sea completamente igual a la respuesta esperada (caso correcto) o si el fragmento extraído contiene a la respuseta esperada y algunos caracteres de más. Dicho caso se considera aprobado porque en esencia el que el documento preprocesado no contenga algunas líneas más de código no afecta considerablemente la extracción focalizada. Como *no aprobado* se entienden los casos en los que la respuesta está incompleta (la respuesta esperada contiene al fragmento de texto obtenido) o que los fragmentos son completamente disjuntos.

5.3.2. Pruebas de Extractor Focalizado

Para probar el extractor focalizado, se tomó al igual que en las pruebas del Preprocesador de texto un conjunto de prueba seleccionado aleatoriamente con uniformidad sobre los años. La muestra es de un

tercio de la población total. Para hacer esto de cada archivo seleccionado, se tomaban hasta 3 unidades de documento según lo explicado en 4.3.2. Dichas unidades fueron vaciadas en un archivo de pruebas que posteriormente era leído por un módulo de pruebas que se encarga de probar el extractor focalizado automáticamente.

Una prueba individual consiste en hacer una búsqueda focalizada sobre un campo presente en una unidad de documento. Una unidad de documento da por lo tanto para hacer tantas pruebas como campos con valor tenga esa unidad. Si el dominio tiene por ejemplo 3 campos y se tiene una unidad de documento con 2 campos, se realizan 2 pruebas para esa unidad: una para cada campo. Las pruebas se hacen iterando entonces sobre cada uno de los campos con valores en la unidad y buscando extraer el valor del mismo utilizando un contexto de extracción dado por los valores de los campos que sí se tienen.

Para simular condiciones en las que no se tienen todos los valores de los campos relacionados con una unidad - esto es, que a pesar de que la unidad contenga valores para varios campos a la hora de hacer la búsqueda focalizada el *contexto de extracción dado por la consulta* no contenga todos esos valores -, los contextos de extracción se construyeron con un subconjunto de los valores de los campos presentes en la unidad de documento. Este subconjunto se determina aleatoriamente: se genera un número al azar y si ese número es mayor que una *probabilidad de tener cada campo* se incluye en el contexto de extracción. De esta manera se evitar suponer para estas pruebas que los contextos de extracción son completos.

Las pruebas se realizaron iterando el valor de la *probabilidad de tener cada campo*. Se tomaron como *probabilidades de tener cada campo* 1; 0.66 y 0.33. Adicionalmente, se iteró también sobre el *Minimum Hit Measure* (según lo definido en 4.3.2) con el objetivo de probar el efecto que tiene la elección de este parámetro en el funcionamiento del extractor focalizado. Los valores sobre los que se realizaron las pruebas fueron: 1; 0.66 y .33.

De esta manera, y resumiendo, las pruebas se realizaron tomando tres unidades de documento de cada documento de cada dominio. Para cada unidad se itera sobre la *probabilidad de tener cada campo* y el *Minimum Hit Measure*. Los resultados fueron posteriormente totalizados por dominio y tabulados.

Capítulo 6

Resultados

A continuación se muestran los resultados de las pruebas realizadas.

6.1. Resultados de las pruebas unitarias sobre el Módulo de Determinación de Origen de la Incompletitud.

6.2. Resultados de las pruebas unitarias sobre el Módulo de Búsqueda Focalizada de Información

A continuación se muestran los resultados de las pruebas unitarias realizadas sobre el Preprocesador de Textos y sobre el Extractor Focalizado de Información.

6.2.1. Resultados de las pruebas unitarias del Preprocesador de Textos

Tabla 6.1: Resultados detallados la evaluación del Preprocesador de Textos

Dominio	**Aprobados**			**No aprobados**		
	Correctos	Con texto de más	**Total aprobados**	Incompletos	Incorrectos	**Total no aprobados**
Designaciones	87.69 %	7.69 %	**95.39 %**	3.07 %	1.54 %	**4.61 %**
Escalafón	80.51 %	12.98 %	**93.6 %**	6.4 %	0 %	**6.4 %**
Jurados de Ascenso	77.55 %	16.32 %	**93.88 %**	0 %	6.12 %	**6.12 %**

6.2.2. Resultados de las pruebas unitarias del Extractor Focalizado

En el apéndice B se muestra la totalidad de los resultados obtenidos de las pruebas unitarias realizadas sobre el Extractor Focalizado. Con el objetivo de seleccionar resultados ilustrativos, para estas pruebas se ha decidido mostrar para cada dominio de información los resultados obtenidos con un valor del Unit Hit Measure de 0.66, mostrando los resultados obtenidos con las probabilidades de campo faltante de 0;

Tabla 6.2: Resultados de la evaluación del Extractor Focalizado - Dominio: Designaciones. UnitHit Measure mínimo:1.0

Campo	Prob.	**P. Aprobadas**			**P. No aprobadas**		
		Correctos	Con texto de más	Aprobados	Incompletos	Incorrectos	No aprobados
calificacion	0.0	74,86 %	9,5 %	**84,36 %**	12,85 %	2,79 %	**15,64 %**
	0.33	65,92 %	7,26 %	**73,18 %**	24,58 %	2,23 %	**26,82 %**
	0.66	39,11 %	5,59 %	**44,69 %**	54,19 %	1,12 %	**55,31 %**
fechaAsignacion	0.0	84,92 %	1,12 %	**86,03 %**	13,41 %	0,56 %	**13,97 %**
	0.33	73,18 %	1,68 %	**74,86 %**	24,58 %	0,56 %	**25,14 %**
	0.66	44,13 %	1,12 %	**45,25 %**	54,75 %	0 %	**54,75 %**
fechaFinal	0.0	96,09 %	1,68 %	**97,77 %**	2,23 %	0 %	**2,23 %**
	0.33	94,41 %	3,91 %	**98,32 %**	1,68 %	0 %	**1,68 %**
	0.66	87,15 %	9,5 %	**96,65 %**	3,35 %	0 %	**3,35 %**
motivo	0.0	88,83 %	7,82 %	**96,65 %**	2,79 %	0,56 %	**3,35 %**
	0.33	78,77 %	13,97 %	**92,74 %**	6,7 %	0,56 %	**7,26 %**
	0.66	63,69 %	17,88 %	**81,56 %**	18,44 %	0 %	**18,44 %**
Nombre	0.0	86,03 %	2,79 %	**88,83 %**	10,61 %	0,56 %	**11,17 %**
	0.33	70,95 %	2,23 %	**73,18 %**	26,82 %	0 %	**26,82 %**
	0.66	47,49 %	2,79 %	**50,28 %**	49,16 %	0,56 %	**49,72 %**

Prob. es la probabilidad de que no se tenga el valor uno de los campos que se utilizan para hacer extracción focalizada.

0.33 y 0.66. El motivo de esto es que se decidió que dicho valor del Unit Hit Measure era representativo de los resultados obtenidos. Se consideró tomar el promedio de los 3 resultados obtenidos según las probabilidades de campo faltante pero se decidió que el efecto de dichas probabilidades era bastante relevante e importante, y por lo tanto conveniente de destacar.

De igual forma, para ilustrar el efecto del valor del Unit Hit Measure se decidió colocar los resultados obtenidos para valores de Unit Hit Measure de 0; 0.33 y 0.66 para el dominio de Designaciones.

TABLAS GO HERE

6.2.3. Análisis de los resultados de las pruebas unitarias del Buscador Focalizado de Información

Los resultados obtenidos varían muchísimo según las tres variables que entran en juego en este experimento: el dominio de información (al cual está asociado una semántica que determina a su vez un extractor especificado por expresiones regulares), el Unit Hit Measure (UHM) y la probabilidad (P) de que cada uno de los campos falten.

Puede verse en primera instancia como algunos campos obtenidos tuvieron una aprobación considerablemente alta: postergado en el Dominio Escalafón obtuvo un 98.46 %, escalafón en el caso de los jurados de ascenso 97,04 % y fecha de Designación en el caso del dominio Designación 98,88 %. De igual forma, hay campos con precisión muchísimos menor: nombre del Miembro Principal Externo en el caso de Jurados de

Tabla 6.3: Resultados de la evaluación del Extractor Focalizado - Dominio: Designaciones. UnitHit Measure mínimo:0.66

Campo	Prob.	**P. Aprobadas**			**P. No aprobadas**		
		Correctos	Con texto de más	Aprobados	Incompletos	Incorrectos	No aprobados
calificacion	0.0	79,33 %	9,5 %	**88,83 %**	8,38 %	2,79 %	**11,17 %**
	0.33	64,8 %	8,38 %	**73,18 %**	24,02 %	2,79 %	**26,82 %**
	0.66	39,66 %	4,47 %	**44,13 %**	54,75 %	1,12 %	**55,87 %**
fechaAsignacion	0.0	83,8 %	2,79 %	**86,59 %**	12,85 %	0,56 %	**13,41 %**
	0.33	74,86 %	2,79 %	**77,65 %**	22,35 %	0 %	**22,35 %**
	0.66	50,28 %	2,23 %	**52,51 %**	46,93 %	0,56 %	**47,49 %**
fechaFinal	0.0	96,09 %	2,79 %	**98,88 %**	1,12 %	0 %	**1,12 %**
	0.33	92,18 %	6,15 %	**98,32 %**	1,68 %	0 %	**1,68 %**
	0.66	85,47 %	10,61 %	**96,09 %**	3,91 %	0 %	**3,91 %**
motivo	0.0	84,92 %	12,85 %	**97,77 %**	1,68 %	0,56 %	**2,23 %**
	0.33	72,63 %	20,11 %	**92,74 %**	7,26 %	0 %	**7,26 %**
	0.66	64,25 %	20,11 %	**84,36 %**	15,64 %	0 %	**15,64 %**
Nombre	0.0	89,39 %	2,79 %	**92,18 %**	7,26 %	0,56 %	**7,82 %**
	0.33	68,16 %	2,79 %	**70,95 %**	28,49 %	0,56 %	**29,05 %**
	0.66	40,78 %	2,23 %	**43,02 %**	56,98 %	0 %	**56,98 %**

Prob. es la probabilidad de que no se tenga el valor uno de los campos que se utilizan para hacer extracción focalizada.

Tabla 6.4: Resultados de la evaluación del Extractor Focalizado - Dominio: Designaciones. UnitHit Measure mínimo:0.33

Campo	Prob.	**P. Aprobadas**			**P. No aprobadas**		
		Correctos	Con texto de más	Aprobados	Incompletos	Incorrectos	No aprobados
calificacion	0.0	80,45 %	9,5 %	**89,94 %**	7,26 %	2,79 %	**10,06 %**
	0.33	63,69 %	8,94 %	**72,63 %**	25,14 %	2,23 %	**27,37 %**
	0.66	40,78 %	5,59 %	**46,37 %**	53,07 %	0,56 %	**53,63 %**
fechaAsignacion	0.0	83,24 %	3,35 %	**86,59 %**	12,85 %	0,56 %	**13,41 %**
	0.33	74,3 %	2,79 %	**77,09 %**	22,35 %	0,56 %	**22,91 %**
	0.66	49,72 %	3,35 %	**53,07 %**	46,93 %	0 %	**46,93 %**
fechaFinal	0.0	82,68 %	16,2 %	**98,88 %**	1,12 %	0 %	**1,12 %**
	0.33	58,1 %	40,78 %	**98,88 %**	1,12 %	0 %	**1,12 %**
	0.66	65,36 %	30,73 %	**96,09 %**	3,91 %	0 %	**3,91 %**
motivo	0.0	59,78 %	37,99 %	**97,77 %**	1,68 %	0,56 %	**2,23 %**
	0.33	49,16 %	44,69 %	**93,85 %**	5,59 %	0,56 %	**6,15 %**
	0.66	53,07 %	31,28 %	**84,36 %**	15,08 %	0,56 %	**15,64 %**
Nombre	0.0	90,5 %	2,79 %	**93,3 %**	6,15 %	0,56 %	**6,7 %**
	0.33	72,07 %	1,68 %	**73,74 %**	25,7 %	0,56 %	**26,26 %**
	0.66	39,11 %	1,68 %	**40,78 %**	58,66 %	0,56 %	**59,22 %**

Prob. es la probabilidad de que no se tenga el valor uno de los campos que se utilizan para hacer extracción focalizada.

Tabla 6.5: Resultados de la evaluación del Extractor Focalizado - Dominio: Escalafon. UnitHit Measure mínimo:0.66

Campo	Prob.	**P. Aprobadas**			**P. No aprobadas**		
		Correctos	Con texto de más	Aprobados	Incompletos	Incorrectos	No aprobados
EsAscendido.Nombre	0.0	92,31 %	1,54 %	**93,85 %**	5,38 %	0,77 %	**6,15 %**
	0.33	76,92 %	1,54 %	**78,46 %**	20,77 %	0,77 %	**21,54 %**
	0.66	43,08 %	0 %	**43,08 %**	56,15 %	0,77 %	**56,92 %**
EsAscendido.NombreTrabajo	0.0	94,62 %	1,54 %	**96,15 %**	2,31 %	1,54 %	**3,85 %**
	0.33	70,77 %	1,54 %	**72,31 %**	26,92 %	0,77 %	**27,69 %**
	0.66	43,08 %	1,54 %	**44,62 %**	53,85 %	1,54 %	**55,38 %**
EsAscendido.escalafon	0.0	97,69 %	0 %	**97,69 %**	2,31 %	0 %	**2,31 %**
	0.33	90,77 %	0 %	**90,77 %**	9,23 %	0 %	**9,23 %**
	0.66	71,54 %	0 %	**71,54 %**	28,46 %	0 %	**28,46 %**
EsAscendido.fecha	0.0	95,38 %	2,31 %	**97,69 %**	2,31 %	0 %	**2,31 %**
	0.33	76,15 %	9,23 %	**85,38 %**	14,62 %	0 %	**14,62 %**
	0.66	60,77 %	8,46 %	**69,23 %**	30,77 %	0 %	**30,77 %**
EsAscendido.postergado	0.0	99,23 %	0 %	**99,23 %**	0,77 %	0 %	**0,77 %**
	0.33	96,15 %	2,31 %	**98,46 %**	1,54 %	0 %	**1,54 %**
	0.66	84,62 %	12,31 %	**96,92 %**	3,08 %	0 %	**3,08 %**

Prob. es la probabilidad de que no se tenga el valor uno de los campos que se utilizan para hacer extracción focalizada.

Tabla 6.6: Resultados de la evaluación del Extractor Focalizado - Dominio: JuradosAscenso. UnitHit Measure mínimo:0.66

Campo	Prob.	**P. Aprobadas**			**P. No aprobadas**		
		Correctos	Con texto de más	Aprobados	Incompletos	Incorrectos	No aprobados
Jurado.MiembroPrincipalExterno	0.0	90,37 %	0 %	**90,37 %**	2,22 %	7,41 %	**9,63 %**
	0.33	88,89 %	0,74 %	**89,63 %**	2,96 %	7,41 %	**10,37 %**
	0.66	75,56 %	7,41 %	**82,96 %**	12,59 %	4,44 %	**17,04 %**
Jurado.MiembroPrincipalInterno	0.0	97,78 %	0 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.33	97,04 %	0 %	**97,04 %**	2,96 %	0 %	**2,96 %**
	0.66	81,48 %	1,48 %	**82,96 %**	17,04 %	0 %	**17,04 %**
Jurado.Presidente	0.0	97,78 %	0 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.33	97,78 %	0 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.66	82,96 %	0 %	**82,96 %**	17,04 %	0 %	**17,04 %**
Jurado.SuplenteExterno	0.0	94,81 %	0 %	**94,81 %**	2,22 %	2,96 %	**5,19 %**
	0.33	91,85 %	2,96 %	**94,81 %**	2,22 %	2,96 %	**5,19 %**
	0.66	77,04 %	6,67 %	**83,7 %**	14,07 %	2,22 %	**16,3 %**
Jurado.SuplenteInterno	0.0	95,56 %	2,22 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.33	95,56 %	2,22 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.66	74,81 %	2,96 %	**77,78 %**	22,22 %	0 %	**22,22 %**
Profesor.Departamento	0.0	97,78 %	0 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.33	97,04 %	0 %	**97,04 %**	2,96 %	0 %	**2,96 %**
	0.66	83,7 %	0 %	**83,7 %**	16,3 %	0 %	**16,3 %**
Profesor.Nombre	0.0	97,78 %	0 %	**97,78 %**	1,48 %	0,74 %	**2,22 %**
	0.33	96,3 %	0 %	**96,3 %**	2,96 %	0,74 %	**3,7 %**
	0.66	77,78 %	0 %	**77,78 %**	21,48 %	0,74 %	**22,22 %**
Trabajo.Escalafon	0.0	98,52 %	0 %	**98,52 %**	1,48 %	0 %	**1,48 %**
	0.33	97,04 %	0 %	**97,04 %**	2,96 %	0 %	**2,96 %**
	0.66	87,41 %	0 %	**87,41 %**	12,59 %	0 %	**12,59 %**
Trabajo.Nombre	0.0	97,78 %	0 %	**97,78 %**	1,48 %	0,74 %	**2,22 %**
	0.33	97,78 %	0 %	**97,78 %**	1,48 %	0,74 %	**2,22 %**
	0.66	77,78 %	0 %	**77,78 %**	21,48 %	0,74 %	**22,22 %**

Prob. es la probabilidad de que no se tenga el valor uno de los campos que se utilizan para hacer extracción focalizada.

Ascenso obtuvó 89,63 %, el nombre del profesor ascendido en el caso de Escalafón 78,46 %, la calificación de la designación (el nombre del cargo y la unidad de la designación) 73,18 %.

Todos estos valores son representativos de cada uno de los 3 dominios y se corresponden a las pruebas hechas con un UHM de 0.66 y P de 0,33. Esto parámetros son considerados representativos y válidos para sacar algunas conclusiones generales. Los resultados obtenidos reflejan muchos de los retos que supone trabajar con expresiones regulares, así como lo útil que es en el ámbito de FIE el contexto de extracción. En lo sucesivo se explicará brevemente los resultados obtenidos y el comportamiento del Extractor Focalizado. Al menos de que se indique lo contrario, al hablar de resultados puntuales se hará referencia a los resultados obtenidos con UHM de 0,66 y P de ,33.

El primer elemento que debe considerarse para entender los resultados es el funcionamiento de las expresiones regulares. Las expresiones regulares utilizadas funcionan especificando delimitadores anteriores y posteriores al valor que se quiere extraer. Dichos delimitadores son muchos casos cadenas de texto estáticas elegidas por un usuario experto luego de analizar el documento y encontrar repeteciones de frases. En muchos casos se especifican en lugar de cadenas de texto clases de caracteres que especifican un conjunto de símbolos que puede repetirse. La mayoría de las veces se busca combinar ambas posibilidades y que los delimitadores estén basados en las cadenas de texto modeladas con clases de caracteres para minimizar los errores de tipeo, de mayúsculas y minúsculas, diferencias en el formato, etcétera.

En todo caso lo relevante en este punto, más allá de entender el proceso que tiene que seguir un usuario experto para configurar el extractor, es entender que a pesar de la riqueza que dan las clases de caracteres, las expresiones regulares pueden considerarse como reglas bastante rígidas y estáticas. Como no son el producto de un proceso de aprendizaje de máquina o de un mecanismo probabilístico, se basan en la configuración que haga el usuario y dependen muchísimo de que la regularidad observada al hacer la configuración sea generalizable a todos los documentos.

Esto es complicado en la medida en la que se haga más amplio el conjunto de documentos sobre los cuales se hace la extracción: a menudo la nomenclatura utilizada, el estilo de redacción, el formato de los documentos, el posicionamiento de la información dentro del documento cambia de un año a otro o inclusive de un mes a otro. La labor del usuario experto es generalizar al máximo las reglas con las que configura el extractor, pero en muchos casos es una tarea cercana a lo imposible.

En muchos casos es imposible generalizar los delimitadores y se depende del conocimiento del usuario experto y de los documentos utilizados para modelar el documento. Por ejemplo, el campo nombre de la designación es uno de los más dificiles para generalizar: el resultado obtenido es uno de los más bajos (70,95 %). El motivo de esto es que los delimitadores utilizados para definir este campo cambian muchísimo

y es poco práctico compilar una lista con todos los posibles nombrse que puede tener una persona.

Por otro lado, hay campos que dada la regularidad de sus valores es fácil de modelar. Es el caso del nombre del escalafón en el dominio de Jurados de trabajo de Ascenso (que tiene 3 valores posibles: titular, asociado y agregado), de las fechas (que se presentan típicamente en formatos dd/MM/aaaa, dd-mm-aaaa), si el ascenso es postergado o no (típicamente se identificamente con una frase del tipo: veredicto desfavorable, veredicto reprobatorio, rechazar dicho trabajo, entre otras). Dichos campos obtuvieron precisiones de 97.04 %, 87.11 (promedio de los 3 campos de tipo fecha presentes en los 3 dominios), 98.46 % respectivamente. Nótese sin embargo, que la precisión de estos campos no es perfecta. En muchos casos una examinación de los resultados puntuales en estos campos indican errores de tipeo que explican los errores del orden del 1-2 %. En otros casos, como en las fechas, existen documentos en los que la fecha se escribe extensivamente en lugar de la sintaxis abreviada (15 de diciembre de mil novecientos noventa y nueve, por ejemplo, en lugar de 15-12-1999).

Otra observación que vale la pena hacer es el efecto que puede tener la estructura del documento y el posicionamiento de los campos dentro del mismo. Esto se puede apreciar en el dominio de designación de jurados de ascenso: los nombres de los miembros del jurado, campos del mismo tipo y que tienen todos el mismo formato (en esencia, el nombre), tienen resultados diferentes según el posicionamiento que tienen dentro del texto. El posicionamiento influye precisamente porque determina la selección de los delimitadores del documento: hay veces que el usuario que construye las expresiones regulares puede beneficiarse de que un campo esté de último en el párrafo o que esté justo después de una palabra clave.

En el caso concreto de los jurados, el resultado promedio de los nombres de los 5 jurados es de 94.8. El resultado del Miembro Principal Exterbi es sin embargo 89,63. Este resultado puede atribuirse a que las expresiones regulares óptimas para este campo fueron construidas esperando que el Miembro Principal Externo sea el último miembro a definirse; en los casos en los que no hay un miembro principal externo el último miembro es el miembro principal interno y por lo tanto se tiene una respuesta equivocada.

Hay también varios casos en los que delimitadores de una unidad de documento tienen errores. Como bien se ha dicho, una unidad de documento es un fragmento de texto que se refiere a una designación, ascenso o jurado de ascenso. Si se tiene más de una unidad en lo que el extractor reconoce como una (por problemas en los delimitadores de las mismas), se pierde toda la información de las unidades adicionales. Esto demuestra lo crítico que es construir las mejores expresiones regulares para delimitar unidades de documento. Quizás sea interesante probar otro mecanismo que no dependa de la existencia de unidades (look ahead, entre otros).

En otro casos, el extractor no permite procesar corrrecciones registradas en los documentos. Es decir,

puede ocurrir que en un acta se especifiquen correcciones sobre la información de una designación, ascenso o jurado que fue anunciado en un acta previa. De cierta manera, se tienen más de una unidad de documento que se refieren a un mismo hecho: la designación, el ascenso o el jurado inicial y las correcciones a los mismos. El extractor es incapaz de diferenciar entre ambos, aunque ambos tengan un UHM alto, y se queda con el primero que consiga. Esto podría solucionarse mejorando el extractor, de forma que pueda identificar cuando una unidad está vinculada a otra unidad presente en el conjunto de los documentos. Esta tarea no puede resolverse exclusivamente escribiendo mejores expresiones regulares: se tiene que introducir una forma de discriminar entre unidades con igual Unit Hit Measure.

De igual forma otro problema asociado al procesamiento por unidades de documento fue encontrado en el dominio de Designaciones. Se encontraron muchísimos casos en los que hay multiples designaciones para una misma persona para un mismo día, y que por lo tanto es imposible diferenciar como unidades diferentes. Esto es, si se tiene un dominio cuya naturaleza hace que los contextos de extracción sean iguales entre unidades de documentos que se refieren a cosas diferentes, se pierde información. Esta hipótesis sirve para explicar los resultados obtenidos en la calificación de la designación: 73,18 %.

Los resultados obtenidos son de esta manera bastante interesantes y la cantidad de información da para sacar muchísimas conclusiones sobre la complejidad que supone la tarea hacer ingeniería de extracción. La tecnología utilizada para hacer la extracción, expresiones regulares, supone retos adicionales y los resultados varían muchísimo según el campo. Es importante recordar que estos resultados son obtenidos en el contexto de documentos semi-estructurados y con una cierta regularidad en su estructura y redacción; es virtualmente imposible pretender hacer extracción en dominios con estilo de redacción más libre salvo para datos muy, muy puntuales. Esto es sin duda alguna uno de los motivos por los cuales las expresiones regulares han sido relegadas en el estudio de las áreas de Procesamiento de Lenguaje Natural y Extracción de Información. Hoy por hoy se apuesta más por métodos estocásticos y aprendizaje de máquina para realizar estas tareas.

Hasta ahora se ha analizado exclusivamente el comportamiento de las expresiones regulares según los dominios y los campos tomando en cuenta si el resultado obtenido es aprobado o no. Los resultados sin embargo pueden ser refinados en dos subcategorías dentro de las categorías aprobado y no aprobado. Dentro de la categoría aprobado se tiene que el resultado sea correcto (que sea un match exacto) o que la respuesta tenga texto de más (la respuesta obtenida contiene a la respuesta correcta). Dentro de la categoría no aprobado se tienen incompletos (la respuesta correcta contiene a la respuesta obtenido) e incompleto (las respuestas obtenidas y correctas son completamente disjuntos).

El motivo por el cual se decidió considerar estas categorías es que se considera ilustrativo e importante para profundizar el entendimiento de los resultados. Por ejemplo, el campo motivo de Designaciones

registró un 20,11 % de respuestas aprobadas que, sin embargo, contienen texto de más. En este caso particular el motivo es que si bien es relativamente fácilmente conseguir el delimitador inicial del motivo de la designación, no es tan fácil conseguir el delimitador final. Por ende sucede a menudo que el extractor consigue el motivo pero captura texto adicional.

De igual forma hay varios campos que registraron porcentajes de incompletitud altos. El nombre del profesor en Designaciones registró un 28,49 % de incompletitud, lo cual quiere decir que si bien se extrajó parte del nombre, no se tuvo completo. El motivo de esto, nuevamente, es que dada la naturaleza del campo y de la posición del mismo en el documento, se hizó muy complicado elegir un delimitador final para el nombre que siempre lo capturara por completo. No es el caso de otros nombres de personas en otros dominios, que sí registraron tasas de acierto más deseables (nombres de los jurados, nombre del profesor que es ascendido, entre otros)

Por último, y no menos importante, la tasa de incorrectitud se refiere a los casos en los que la respuesta obtenida no tiene absolutamente nada que ver con la respuesta correcta (no contiene ni es contenida una por la otra). En casí todos los casos la proporción de respuestas es despreciable: en la mayoría de los casos es 0 (11 de 19), el promedio de la proporción de incorrectitud de todos los campos 0,87 % y sólo en un campo pasa de 3 % (el nombre del miembro principal externo, con un 7,41 %). La existencia de estas respuestas incorrectas puede atribuirse a problemas en la selección de los delimitadores.

Queda en este punto comentar sobre el efecto que tiene la probabilidad de que falte un campo y el Hit Measure en los resultados obtenidos. Cómo se ha dicho anteriormente la probabilidad de que falte un campo es un elemento introducido en los experimentos para modelar el efecto que tiene la riqueza que tiene el contexto de extracción a la hora de hacer una búsqueda focalizada. Los archivos que se utilizaron para generar las pruebas contienen toda la información que se puede extraer de los documentos; sin embargo a la hora de hacer las pruebas, al incorporar el valor de cada campo se genera un número aleatorio; si es mayor que la probabilidad de la prueba se incorpora al contexto, de lo contrario no se incorpora. Esto permite simular condiciones normales de una extracción focalizada en la que no se tienen todos los valores de los campos.

El efecto de la probabilidad en los resultados obtenidos es bastante claro: se puede observar que conforme aumenta la probabilidad de que falte un campo (esto es, que el contexto de extracción sea menos rico) disminuye la proporción de resultados aprobados. Este comportamiento se puede observar en practicamente todos los campos y dominios. La disminución de la tasa de aprobación varía sin embargo bastante de acuerdo al campo: en algunos campos es del orden de las décimas mientras que en algunos otros es del orden de las decenas. Esto puede deberse a que existen campos que son más relevantes para poder discriminar entre unidades de documento. Si toca el caso de que el campo que se esté buscando es

importante para identificar cada unidad, el efecto de que el contexto esté incompleto puede ser mayor.

Por último el UHM es una medida de qué tan exigente es el extractor focalizado al seleccionar unidades de documento. Esto es, mide que porcentaje de campos contenidos en el contexto de extracción están presentes en cada unidad de documento: 1.0 indica que todos los valores del contexto están en la unidad, .66 y .33 que el 66 % y el 33 % están. En general en la medida en la que se flexibiliza la exigencia del extractor con el UHM, aumenta el número de resultados aprobados. El motivo de esto es que relajar la exigencia sólo permite que en caso de no tener mejores unidades se procesen unidades con una menor probabilidad de que se refieran al contexto de extracción. El extractor, como se ha dicho, ordena las unidades encontradas de acuerdo al unit hit measure y busca extraer primero el valor del campo faltante de las unidades con mayor UHM.

La importante del Unit Hit Measure es que de no tener exigencias sobre el mínimo valor de cada unidad se pueden tener resultados completamente incorrectos si el extractor no ubica las unidades de documento correctas y pasa a analizar unidades con menor UHM.

A manera de conclusión, el análisis aquí busca ilustrar, más que desmenuzar toda la información que se puede extraer de estas pruebas, los principales hallazgos producto de la utilización del extractor. Los hallazgos más importantes son que la utilización de expresiones regulares depende, como podría esperarse, fuertemente de la regularidad de un documento en su estructura y estilo de redacción, que está condicionada a errores humanos de tipeo y que depende de los delimitadores que un usuario experto pueda definir. De igual forma, depende fuertemente de cada uno de los campos y de la forma como se encuentran esos campos en el texto. Influye también la posición relativa que tengan cada uno de ellos.

De igual manera, la existencia de información previa (el contexto de extracción, que es precisamente la principal ventaja de la extracción focalizada) incide determinantemente en el hallazgo de información faltante en la ontología.

Conclusiones y recomendaciones

El presente proyecto tuvo como objetivo diseñar e implementar un mecanismo de extracción focalizada según lo especificado en la Arquitectura para la Interrogación de Documentos (DIA) propuesta por Abad Mota en [1]. La solución propuesta e implementada soluciona el problema de la determinación de orígen de incompletitud por desconocimiento de valores de atributos, la construcción de un contexto de extracción basado en valores de atributos relacionados con el atributo faltante y la búsqueda focalizada mediante un extractor basado en expresiones regulares.

Los resultados obtenidos son:
A manera de conclusión, es pertinente resumir algunas reflexiones derivadas de este trabajo que pueden ser de interés para profundizar la investigación en esta área.

Primeramente, la premisa fundamental sobre la cual opera DIA al definir la extracción focalizada es que existe información presente en la ontología y en la base de datos con la cual se puede realizar una segunda extracción sobre un conjunto de documentos para descubrir información faltante. Si bien este proyecto ha sido un primer paso importante en el razonamiento sobre la extracción focalizada, existen varias interrogantes que deben ser respondidas en proyectos que busquen probar cosas más específicas con tecnologías más poderosas.

En primer lugar, algo muy importante sobre lo cual se debe razonar para ahondar en el concepto de extracción focalizada es qué metadatos se pueden derivar o tener presentes en la ontología para guiar la extracción focalizada. Esto es quizás la principal fortaleza que tiene este mecanismo y merece muchísima atención. En este trabajo se utilizaron valores de atributos conocidos para realizar la búsqueda focalizada y se obtuvieron resultados satisfactorios, dada la naturaleza de los dominios de información con los que se trabajó y otras consideraciones que se han mencionado anteriormente. Sin embargo, este modelo es probablemente simple y se puede asociar a casos en los que los documentos contengan unidades de información, según lo definido anteriormente, que puedan ser explotados por extractores. Una posible profundización en el desarrollo de DIA es el estudio de nuevos metadatos e información que se pueda incluir en lo que se definió como un contexto de extracción en este trabajo.

Esto lleva a otra segunda conclusión de altísima relevancia. La tarea de la extracción (y el contexto de extracción) depende de las tecnologías existentes para ello. En este trabajo se utilizaron expresiones regulares. Se mencionó, sin embargo, que la investigación en esta área ha estado guiada en los últimos años por la utilización de métodos estocásticos y de inteligencia artificial que han demostrado mejores resultados. La utilización de dichos mecanismos escapa por completo del alcance definido para este proyecto, concebido como un primer paso en el área de extracción focalizada. Sin embargo, un reto que queda pendiente es razonar en la construcción de extractores de información con las últimas y las mejores tecnologías existentes a partir de los metadatos que se puedan extraer de la ontología.

Otro resultado derivado del trabajo es lo importante que supone la tarea de hacer *ingeniería* de documentos para la extracción focalizada. Ademas de construir el extractor de información y el contexto de extracción, es necesario realizar tareas adicionales cuando la información está presente en más de un tipo de documento, cuando los documentos no son los documentos originales, cuando se quiere trabajar con el World Wide Web, cuando los estilos y tipos de documentos son heterógeneos en su estilo y forma de redacción o simplemente poco estructurados, entre otros.

Además de lo ya expuesto anteriormente, es importante tomar en cuenta es que se decidió trabajar con sólo un tipo de origen de incompletitud: la incompletitud por el desconocimiento de algunos valores de atributos en la ontología. Dicha decisión fue tomada para acotar y delimitar los alcances de este proyecto. Sin embargo queda pendiente razonar sobre incompletitudes por inexistencia de una instancia en la ontología o por el desconocimiento en la interrelación entre dos conceptos.

Un aspecto que quedó planteado en el diseño de la solución pero que no fue implementado es el manejo de condiciones en una consulta que consistan en conjunciones de cadenas de disjunciones. Un primer paso para continuar el desarrollo en el área de la Extracción Focalizada de información consiste en implementar esas consideraciones.

De forma que en conclusión, este proyecto de grado puede interpretarse como un primer paso en un tema como la extracción focalizada que resulta bastante prometedor pero que aún tiene muchísimas interrogantes y áreas de crecimiento. En este último capítulo se ha buscado resumir las principales áreas. Es de alto interés evaluar los resultados obtenidos a la luz de tecnologías y avances científicos que permitan respondar las interrogantes planteadas.

Bibliografía

[1] Abad Mota Soraya, "Document interrogation: Architecture, information extraction and approximate answers", 2009.

[2] Eduardo José Ruiz Irigoyen, "Aprendizaje supervisado de sistemas de extracción de información para texto semi-estructurado", Master's thesis, Universidad Simón Bolívar. Trabajo de Grado. Magister en Ciencias de la Computación, 2006.

[3] Ruiz Eduardo, "Extracción de información en texto semi-estructurado", 2004.

[4] APAZA G. and MIRISOLA R., "Implementación del lenguaje odil de interrogación de documentos basado en una ontología", 2005.

[5] Leire Ituarte Pérez, "Implementación del lenguaje odil de interrogación de documentos basado en una ontología", 2004.

[6] Aggarwal and Fellow, "A survey of uncertain data algorithms and applications", *IEEE*, vol. 21, pp. 609–623, 2009.

[7] D. KALASHNIKOV R. CHENG and S.PRABHAKAR, "Evaluating probabilistic queries over imprecise data", *Proc ACM SIGMOD*, 2003.

[8] J. CHEN R. CHENG and Xie X., "Cleaning uncertain data with quality guarantees", *Proceedings of the VLDB Endowmen*, vol. 1,1, pp. 722–735, 2008.

[9] Lipski Witold, "On databases with incomplete information", *Journal of ACM*, vol. 8,1, 1981.

[10] In-ho Kang and Kim GilChang, "Query type classification for web document retrieval", *Proc Sigir '03*, 2003.

[11] Segoufin Rocquenco and Viano, "Representing and querying xml with incomplete information", *PODS'01*, 2001.

[12] Ling Liu and M. Tamer Özsu, *Encyclopedia of Database Systems*, Springer-Verlag, 2009.

[13] Abad S. and Helman P., *ODIL: Ontology-based Document Interrogation Language*, 2004.

[14] Eric W Weisstein, "Conjunctive normal form", `http://mathworld.wolfram.com/ConjunctiveNormalForm.html`, MathWorld–A Wolfram Web Resource.

[15] Jason Eisner, "How to convert a formula to cnf", `http://cs.jhu.edu/~jason/tutorials/convert-to-CNF`.

Apéndices

Apéndice A

Ejemplos de archivos de configuración XML de Preprocesador de Texto y Extractor Focalizado

A continuación se presentan dos archivo de configuración del Preprocesador de Texto y del Extractor Focalizado del Buscador Focalizado de Información.

Código A.1: Archivo de configuración del Preprocesador de Texto con Dominio Designaciones

```
<?xml version="1.0" encoding="UTF-8"?>

<Filesources>

    <Filesource>

        <Name>Designaciones</Name>

        <InputFilePath>/home/frandres/Eclipse/workspace/ExtractionModule/tests/
            Designaciones/rawDocuments/</InputFilePath>

        <OutputFilePath>/home/frandres/Eclipse/workspace/ExtractionModule/tests/
            Designaciones/preProcessedDocuments/</OutputFilePath>

        <RegExps>

            <RegExp>
                <Priority>1</Priority>
                <String>De acuerdo al numeral 7 del articulo 16(.*?)[\^0-9 ][23(?:IV)(?:
                    III)(?:II)][\.\t].*</String>
            </RegExp>

            <RegExp>
                <Priority>2</Priority>
                <String>.*(?:[Ii][Nn][Ff][Oo][Rr][Mm][Ee] [Dd][Ee][Ll] [Rr][Ee][Cc][Tt][
                    Oo][Rr]).*?([Dd][Ee][Ss][Ii][Gg][Nn][Aa][Cc][Ii][Oo\u00f3][Nn](?:[Ee
                    ][Ss]){0,1}.*?)[^0-9 ][23(?:IV)(?:III)(?:II)][\.\t].*</String>
            </RegExp>

        </RegExps>

        <readAllFiles>1</readAllFiles>

        <outputExtension>.txt</outputExtension>

    </Filesource>
</Filesources>
```

Código A.2: Fragmento del Archivo de configuración del Extractor Focalizado de Texto con Dominio Designaciones

```
<Configuration>
    <UnitRegExps>
        <UnitRegExp>^(.+?)$</UnitRegExp>
    </UnitRegExps>


    <MinimumHitRatio>.5</MinimumHitRatio>
    <DocumentsFilePath>/home/frandres/Eclipse/workspace/ExtractionModule/tests/
        Designaciones/preProcessedDocuments/</DocumentsFilePath>
    <FieldDescriptors>

        <FieldDescriptor>
            <FieldName>EsAsignado.calificacion</FieldName>
            <SpecificRegExps>
                <SpecificRegExp priority="1" >([Jj]ef.*?)(?:(?:a\s+partir\s+del{0,1}
                    \s)|(?:desde)|(?:hasta)|(?:por motivo)|(?:en sustitu)|(?:del\s\d)
                    |(?:al\s\d)|(?:\s+entre\s+el\s+\d)|(?:por\s+el\s+periodo\s+)).*</
                    SpecificRegExp>
                <SpecificRegExp priority="2" >([Cc]oordinador.*?)(?:(?:a\s+partir\s
                    +del{0,1}\s)|(?:desde)|(?:hasta)|(?:por motivo)|(?:en sustitu)
                    |(?:del\s\d)|(?:al\s\d)|(?:\s+entre\s+el\s+\d)|(?:por\s+el\s+
                    periodo\s+)).*</SpecificRegExp>

                <SpecificRegExp priority="13" >([Jj]ef.*)</SpecificRegExp>
                <SpecificRegExp priority="14" >([Cc]oordinador.*)</SpecificRegExp>

            </SpecificRegExps>
            <Weight>1</Weight>
            <isDate>0</isDate>
        </FieldDescriptor>
    </FieldDescriptors>

</Configuration>
```

Apéndice B

Resultados de las pruebas unitarias realizadas sobre el Buscador Focalizado de Información

Tabla B.1: Resultados de la evaluación del Extractor Focalizado - Dominio: Designaciones. UnitHit Measure mínimo:1.0

Campo	Prob. Campo Faltante	**P. Aprobadas**			**P. No aprobadas**		
		Correctos	Con texto de más	Aprobados	Incompletos	Incorrectos	No aprobados
EsAsignado.calificacion	0.0	74,86 %	9,5 %	**84,36 %**	12,85 %	2,79 %	**15,64 %**
	0.33	65,92 %	7,26 %	**73,18 %**	24,58 %	2,23 %	**26,82 %**
	0.66	39,11 %	5,59 %	**44,69 %**	54,19 %	1,12 %	**55,31 %**
EsAsignado.fechaAsignacion	0.0	84,92 %	1,12 %	**86,03 %**	13,41 %	0,56 %	**13,97 %**
	0.33	73,18 %	1,68 %	**74,86 %**	24,58 %	0,56 %	**25,14 %**
	0.66	44,13 %	1,12 %	**45,25 %**	54,75 %	0 %	**54,75 %**
EsAsignado.fechaFinal	0.0	96,09 %	1,68 %	**97,77 %**	2,23 %	0 %	**2,23 %**
	0.33	94,41 %	3,91 %	**98,32 %**	1,68 %	0 %	**1,68 %**
	0.66	87,15 %	9,5 %	**96,65 %**	3,35 %	0 %	**3,35 %**
EsAsignado.motivo	0.0	88,83 %	7,82 %	**96,65 %**	2,79 %	0,56 %	**3,35 %**
	0.33	78,77 %	13,97 %	**92,74 %**	6,7 %	0,56 %	**7,26 %**
	0.66	63,69 %	17,88 %	**81,56 %**	18,44 %	0 %	**18,44 %**
Profesor.Nombre	0.0	86,03 %	2,79 %	**88,83 %**	10,61 %	0,56 %	**11,17 %**
	0.33	70,95 %	2,23 %	**73,18 %**	26,82 %	0 %	**26,82 %**
	0.66	47,49 %	2,79 %	**50,28 %**	49,16 %	0,56 %	**49,72 %**

Prob. Campo Faltante es la probabilidad de que no se tenga el valor uno de los campos que se utilizan para hacer extracción focalizada.

Tabla B.2: Resultados de la evaluación del Extractor Focalizado - Dominio: Designaciones. UnitHit Measure mínimo:0.66

Campo	Prob. Campo Faltante	**P. Aprobadas**			**P. No aprobadas**		
		Correctos	Con texto de más	Aprobados	Incompletos	Incorrectos	No aprobados
EsAsignado.calificacion	0.0	79,33 %	9,5 %	**88,83 %**	8,38 %	2,79 %	**11,17 %**
	0.33	64,8 %	8,38 %	**73,18 %**	24,02 %	2,79 %	**26,82 %**
	0.66	39,66 %	4,47 %	**44,13 %**	54,75 %	1,12 %	**55,87 %**
EsAsignado.fechaAsignacion	0.0	83,8 %	2,79 %	**86,59 %**	12,85 %	0,56 %	**13,41 %**
	0.33	74,86 %	2,79 %	**77,65 %**	22,35 %	0 %	**22,35 %**
	0.66	50,28 %	2,23 %	**52,51 %**	46,93 %	0,56 %	**47,49 %**
EsAsignado.fechaFinal	0.0	96,09 %	2,79 %	**98,88 %**	1,12 %	0 %	**1,12 %**
	0.33	92,18 %	6,15 %	**98,32 %**	1,68 %	0 %	**1,68 %**
	0.66	85,47 %	10,61 %	**96,09 %**	3,91 %	0 %	**3,91 %**
EsAsignado.motivo	0.0	84,92 %	12,85 %	**97,77 %**	1,68 %	0,56 %	**2,23 %**
	0.33	72,63 %	20,11 %	**92,74 %**	7,26 %	0 %	**7,26 %**
	0.66	64,25 %	20,11 %	**84,36 %**	15,64 %	0 %	**15,64 %**
Profesor.Nombre	0.0	89,39 %	2,79 %	**92,18 %**	7,26 %	0,56 %	**7,82 %**
	0.33	68,16 %	2,79 %	**70,95 %**	28,49 %	0,56 %	**29,05 %**
	0.66	40,78 %	2,23 %	**43,02 %**	56,98 %	0 %	**56,98 %**

Prob. Campo Faltante es la probabilidad de que no se tenga el valor uno de los campos que se utilizan para hacer extracción focalizada.

Tabla B.3: Resultados de la evaluación del Extractor Focalizado - Dominio: Designaciones. UnitHit Measure mínimo:0.33

Campo	Prob. Campo Faltante	**P. Aprobadas**			**P. No aprobadas**		
		Correctos	Con texto de más	Aprobados	Incompletos	Incorrectos	No aprobados
EsAsignado.calificacion	0.0	80,45 %	9,5 %	**89,94 %**	7,26 %	2,79 %	**10,06 %**
	0.33	63,69 %	8,94 %	**72,63 %**	25,14 %	2,23 %	**27,37 %**
	0.66	40,78 %	5,59 %	**46,37 %**	53,07 %	0,56 %	**53,63 %**
EsAsignado.fechaAsignacion	0.0	83,24 %	3,35 %	**86,59 %**	12,85 %	0,56 %	**13,41 %**
	0.33	74,3 %	2,79 %	**77,09 %**	22,35 %	0,56 %	**22,91 %**
	0.66	49,72 %	3,35 %	**53,07 %**	46,93 %	0 %	**46,93 %**
EsAsignado.fechaFinal	0.0	82,68 %	16,2 %	**98,88 %**	1,12 %	0 %	**1,12 %**
	0.33	58,1 %	40,78 %	**98,88 %**	1,12 %	0 %	**1,12 %**
	0.66	65,36 %	30,73 %	**96,09 %**	3,91 %	0 %	**3,91 %**
EsAsignado.motivo	0.0	59,78 %	37,99 %	**97,77 %**	1,68 %	0,56 %	**2,23 %**
	0.33	49,16 %	44,69 %	**93,85 %**	5,59 %	0,56 %	**6,15 %**
	0.66	53,07 %	31,28 %	**84,36 %**	15,08 %	0,56 %	**15,64 %**
Profesor.Nombre	0.0	90,5 %	2,79 %	**93,3 %**	6,15 %	0,56 %	**6,7 %**
	0.33	72,07 %	1,68 %	**73,74 %**	25,7 %	0,56 %	**26,26 %**
	0.66	39,11 %	1,68 %	**40,78 %**	58,66 %	0,56 %	**59,22 %**

Prob. Campo Faltante es la probabilidad de que no se tenga el valor uno de los campos que se utilizan para hacer extracción focalizada.

Tabla B.4: Resultados de la evaluación del Extractor Focalizado - Dominio: Escalafon. UnitHit Measure mínimo:1.0

Campo	Prob. Campo Faltante	**P. Aprobadas**			**P. No aprobadas**		
		Correctos	Con texto de más	Aprobados	Incompletos	Incorrectos	No aprobados
EsAscendido.Nombre	0.0	92,31 %	1,54 %	**93,85 %**	5,38 %	0,77 %	**6,15 %**
	0.33	77,69 %	0 %	**77,69 %**	21,54 %	0,77 %	**22,31 %**
	0.66	43,85 %	1,54 %	**45,38 %**	53,85 %	0,77 %	**54,62 %**
EsAscendido.NombreTrabajo	0.0	94,62 %	1,54 %	**96,15 %**	3,08 %	0,77 %	**3,85 %**
	0.33	73,85 %	1,54 %	**75,38 %**	23,85 %	0,77 %	**24,62 %**
	0.66	42,31 %	0,77 %	**43,08 %**	56,15 %	0,77 %	**56,92 %**
EsAscendido.escalafon	0.0	97,69 %	0 %	**97,69 %**	2,31 %	0 %	**2,31 %**
	0.33	90 %	0 %	**90 %**	10 %	0 %	**10 %**
	0.66	65,38 %	0 %	**65,38 %**	34,62 %	0 %	**34,62 %**
EsAscendido.fecha	0.0	96,15 %	1,54 %	**97,69 %**	2,31 %	0 %	**2,31 %**
	0.33	84,62 %	4,62 %	**89,23 %**	10,77 %	0 %	**10,77 %**
	0.66	54,62 %	6,15 %	**60,77 %**	39,23 %	0 %	**39,23 %**
EsAscendido.postergado	0.0	99,23 %	0 %	**99,23 %**	0,77 %	0 %	**0,77 %**
	0.33	94,62 %	3,85 %	**98,46 %**	1,54 %	0 %	**1,54 %**
	0.66	89,23 %	7,69 %	**96,92 %**	3,08 %	0 %	**3,08 %**

Prob. Campo Faltante es la probabilidad de que no se tenga el valor uno de los campos que se utilizan para hacer extracción focalizada.

Tabla B.5: Resultados de la evaluación del Extractor Focalizado - Dominio: Escalafon. UnitHit Measure mínimo:0.66

Campo	Prob. Campo Faltante	**P. Aprobadas**			**P. No aprobadas**		
		Correctos	Con texto de más	Aprobados	Incompletos	Incorrectos	No aprobados
EsAscendido.Nombre	0.0	92,31 %	1,54 %	**93,85 %**	5,38 %	0,77 %	**6,15 %**
	0.33	76,92 %	1,54 %	**78,46 %**	20,77 %	0,77 %	**21,54 %**
	0.66	43,08 %	0 %	**43,08 %**	56,15 %	0,77 %	**56,92 %**
EsAscendido.NombreTrabajo	0.0	94,62 %	1,54 %	**96,15 %**	2,31 %	1,54 %	**3,85 %**
	0.33	70,77 %	1,54 %	**72,31 %**	26,92 %	0,77 %	**27,69 %**
	0.66	43,08 %	1,54 %	**44,62 %**	53,85 %	1,54 %	**55,38 %**
EsAscendido.escalafon	0.0	97,69 %	0 %	**97,69 %**	2,31 %	0 %	**2,31 %**
	0.33	90,77 %	0 %	**90,77 %**	9,23 %	0 %	**9,23 %**
	0.66	71,54 %	0 %	**71,54 %**	28,46 %	0 %	**28,46 %**
EsAscendido.fecha	0.0	95,38 %	2,31 %	**97,69 %**	2,31 %	0 %	**2,31 %**
	0.33	76,15 %	9,23 %	**85,38 %**	14,62 %	0 %	**14,62 %**
	0.66	60,77 %	8,46 %	**69,23 %**	30,77 %	0 %	**30,77 %**
EsAscendido.postergado	0.0	99,23 %	0 %	**99,23 %**	0,77 %	0 %	**0,77 %**
	0.33	96,15 %	2,31 %	**98,46 %**	1,54 %	0 %	**1,54 %**
	0.66	84,62 %	12,31 %	**96,92 %**	3,08 %	0 %	**3,08 %**

Prob. Campo Faltante es la probabilidad de que no se tenga el valor uno de los campos que se utilizan para hacer extracción focalizada.

Tabla B.6: Resultados de la evaluación del Extractor Focalizado - Dominio: Escalafon. UnitHit Measure mínimo:0.33

Campo	Prob. Campo Faltante	**P. Aprobadas**			**P. No aprobadas**		
		Correctos	Con texto de más	Aprobados	Incompletos	Incorrectos	No aprobados
EsAscendido.Nombre	0.0	92,31 %	1,54 %	**93,85 %**	5,38 %	0,77 %	**6,15 %**
	0.33	70,77 %	1,54 %	**72,31 %**	26,92 %	0,77 %	**27,69 %**
	0.66	41,54 %	0,77 %	**42,31 %**	56,92 %	0,77 %	**57,69 %**
EsAscendido.NombreTrabajo	0.0	94,62 %	1,54 %	**96,15 %**	2,31 %	1,54 %	**3,85 %**
	0.33	72,31 %	1,54 %	**73,85 %**	24,62 %	1,54 %	**26,15 %**
	0.66	41,54 %	0,77 %	**42,31 %**	56,15 %	1,54 %	**57,69 %**
EsAscendido.escalafon	0.0	99,23 %	0 %	**99,23 %**	0,77 %	0 %	**0,77 %**
	0.33	92,31 %	0 %	**92,31 %**	7,69 %	0 %	**7,69 %**
	0.66	70,77 %	0 %	**70,77 %**	29,23 %	0 %	**29,23 %**
EsAscendido.fecha	0.0	80 %	17,69 %	**97,69 %**	2,31 %	0 %	**2,31 %**
	0.33	70,77 %	16,92 %	**87,69 %**	12,31 %	0 %	**12,31 %**
	0.66	56,92 %	14,62 %	**71,54 %**	28,46 %	0 %	**28,46 %**
EsAscendido.postergado	0.0	83,08 %	16,15 %	**99,23 %**	0,77 %	0 %	**0,77 %**
	0.33	52,31 %	46,92 %	**99,23 %**	0,77 %	0 %	**0,77 %**
	0.66	66,15 %	31,54 %	**97,69 %**	2,31 %	0 %	**2,31 %**

Prob. Campo Faltante es la probabilidad de que no se tenga el valor uno de los campos que se utilizan para hacer extracción focalizada.

Tabla B.7: Resultados de la evaluación del Extractor Focalizado - Dominio: JuradosAscenso. UnitHit Measure mínimo:1.0

Campo	Prob. Campo Faltante	**P. Aprobadas**			**P. No aprobadas**		
		Correctos	Con texto de más	Aprobados	Incompletos	Incorrectos	No aprobados
Jurado.MiembroPrincipalExterno	0.0	89,63 %	0 %	**89,63 %**	5,93 %	4,44 %	**10,37 %**
	0.33	89,63 %	0 %	**89,63 %**	5,93 %	4,44 %	**10,37 %**
	0.66	77,78 %	2,22 %	**80 %**	14,81 %	5,19 %	**20 %**
Jurado.MiembroPrincipalInterno	0.0	97,04 %	0 %	**97,04 %**	2,96 %	0 %	**2,96 %**
	0.33	97,04 %	0 %	**97,04 %**	2,96 %	0 %	**2,96 %**
	0.66	81,48 %	0,74 %	**82,22 %**	17,78 %	0 %	**17,78 %**
Jurado.Presidente	0.0	89,63 %	0 %	**89,63 %**	10,37 %	0 %	**10,37 %**
	0.33	91,11 %	0 %	**91,11 %**	8,89 %	0 %	**8,89 %**
	0.66	81,48 %	0 %	**81,48 %**	18,52 %	0 %	**18,52 %**
Jurado.SuplenteExterno	0.0	89,63 %	0 %	**89,63 %**	10,37 %	0 %	**10,37 %**
	0.33	92,59 %	0 %	**92,59 %**	6,67 %	0,74 %	**7,41 %**
	0.66	79,26 %	2,96 %	**82,22 %**	16,3 %	1,48 %	**17,78 %**
Jurado.SuplenteInterno	0.0	93,33 %	0 %	**93,33 %**	6,67 %	0 %	**6,67 %**
	0.33	94,07 %	0 %	**94,07 %**	5,93 %	0 %	**5,93 %**
	0.66	82,22 %	0 %	**82,22 %**	17,78 %	0 %	**17,78 %**
Profesor.Departamento	0.0	88,89 %	0 %	**88,89 %**	11,11 %	0 %	**11,11 %**
	0.33	91,85 %	0 %	**91,85 %**	8,15 %	0 %	**8,15 %**
	0.66	88,15 %	0 %	**88,15 %**	11,85 %	0 %	**11,85 %**
Profesor.Nombre	0.0	88,89 %	0 %	**88,89 %**	10,37 %	0,74 %	**11,11 %**
	0.33	88,89 %	0 %	**88,89 %**	10,37 %	0,74 %	**11,11 %**
	0.66	80,74 %	0 %	**80,74 %**	18,52 %	0,74 %	**19,26 %**
Trabajo.Escalafon	0.0	89,63 %	0 %	**89,63 %**	10,37 %	0 %	**10,37 %**
	0.33	91,85 %	0 %	**91,85 %**	8,15 %	0 %	**8,15 %**
	0.66	85,19 %	0 %	**85,19 %**	14,81 %	0 %	**14,81 %**
Trabajo.Nombre	0.0	89,63 %	0 %	**89,63 %**	9,63 %	0,74 %	**10,37 %**
	0.33	91,11 %	0 %	**91,11 %**	8,15 %	0,74 %	**8,89 %**
	0.66	80,74 %	0 %	**80,74 %**	19,26 %	0 %	**19,26 %**

Prob. Campo Faltante es la probabilidad de que no se tenga el valor uno de los campos que se utilizan para hacer extracción focalizada.

Tabla B.8: Resultados de la evaluación del Extractor Focalizado - Dominio: JuradosAscenso. UnitHit Measure mínimo:0.66

Campo	Prob. Campo Faltante	**P. Aprobadas**			**P. No aprobadas**		
		Correctos	Con texto de más	Aprobados	Incompletos	Incorrectos	No aprobados
Jurado.MiembroPrincipalExterno	0.0	90,37 %	0 %	**90,37 %**	2,22 %	7,41 %	**9,63 %**
	0.33	88,89 %	0,74 %	**89,63 %**	2,96 %	7,41 %	**10,37 %**
	0.66	75,56 %	7,41 %	**82,96 %**	12,59 %	4,44 %	**17,04 %**
Jurado.MiembroPrincipalInterno	0.0	97,78 %	0 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.33	97,04 %	0 %	**97,04 %**	2,96 %	0 %	**2,96 %**
	0.66	81,48 %	1,48 %	**82,96 %**	17,04 %	0 %	**17,04 %**
Jurado.Presidente	0.0	97,78 %	0 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.33	97,78 %	0 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.66	82,96 %	0 %	**82,96 %**	17,04 %	0 %	**17,04 %**
Jurado.SuplenteExterno	0.0	94,81 %	0 %	**94,81 %**	2,22 %	2,96 %	**5,19 %**
	0.33	91,85 %	2,96 %	**94,81 %**	2,22 %	2,96 %	**5,19 %**
	0.66	77,04 %	6,67 %	**83,7 %**	14,07 %	2,22 %	**16,3 %**
Jurado.SuplenteInterno	0.0	95,56 %	2,22 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.33	95,56 %	2,22 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.66	74,81 %	2,96 %	**77,78 %**	22,22 %	0 %	**22,22 %**
Profesor.Departamento	0.0	97,78 %	0 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.33	97,04 %	0 %	**97,04 %**	2,96 %	0 %	**2,96 %**
	0.66	83,7 %	0 %	**83,7 %**	16,3 %	0 %	**16,3 %**
Profesor.Nombre	0.0	97,78 %	0 %	**97,78 %**	1,48 %	0,74 %	**2,22 %**
	0.33	96,3 %	0 %	**96,3 %**	2,96 %	0,74 %	**3,7 %**
	0.66	77,78 %	0 %	**77,78 %**	21,48 %	0,74 %	**22,22 %**
Trabajo.Escalafon	0.0	98,52 %	0 %	**98,52 %**	1,48 %	0 %	**1,48 %**
	0.33	97,04 %	0 %	**97,04 %**	2,96 %	0 %	**2,96 %**
	0.66	87,41 %	0 %	**87,41 %**	12,59 %	0 %	**12,59 %**
Trabajo.Nombre	0.0	97,78 %	0 %	**97,78 %**	1,48 %	0,74 %	**2,22 %**
	0.33	97,78 %	0 %	**97,78 %**	1,48 %	0,74 %	**2,22 %**
	0.66	77,78 %	0 %	**77,78 %**	21,48 %	0,74 %	**22,22 %**

Prob. Campo Faltante es la probabilidad de que no se tenga el valor uno de los campos que se utilizan para hacer extracción focalizada.

Tabla B.9: Resultados de la evaluación del Extractor Focalizado - Dominio: JuradosAscenso. UnitHit Measure mínimo:0.33

Campo	Prob. Campo Faltante	**P. Aprobadas**			**P. No aprobadas**		
		Correctos	Con texto de más	Aprobados	Incompletos	Incorrectos	No aprobados
Jurado.MiembroPrincipalExterno	0.0	67,41 %	22,96 %	**90,37 %**	2,22 %	7,41 %	**9,63 %**
	0.33	70,37 %	20 %	**90,37 %**	2,22 %	7,41 %	**9,63 %**
	0.66	62,22 %	17,04 %	**79,26 %**	13,33 %	7,41 %	**20,74 %**
Jurado.MiembroPrincipalInterno	0.0	91,85 %	5,93 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.33	91,11 %	5,19 %	**96,3 %**	3,7 %	0 %	**3,7 %**
	0.66	77,78 %	4,44 %	**82,22 %**	17,78 %	0 %	**17,78 %**
Jurado.Presidente	0.0	97,78 %	0 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.33	97,04 %	0 %	**97,04 %**	2,96 %	0 %	**2,96 %**
	0.66	82,96 %	0 %	**82,96 %**	17,04 %	0 %	**17,04 %**
Jurado.SuplenteExterno	0.0	71,85 %	22,96 %	**94,81 %**	2,22 %	2,96 %	**5,19 %**
	0.33	74,81 %	20 %	**94,81 %**	2,22 %	2,96 %	**5,19 %**
	0.66	68,89 %	19,26 %	**88,15 %**	10,37 %	1,48 %	**11,85 %**
Jurado.SuplenteInterno	0.0	93,33 %	4,44 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.33	93,33 %	4,44 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.66	82,96 %	2,96 %	**85,93 %**	14,07 %	0 %	**14,07 %**
Profesor.Departamento	0.0	98,52 %	0,74 %	**99,26 %**	0,74 %	0 %	**0,74 %**
	0.33	98,52 %	0 %	**98,52 %**	1,48 %	0 %	**1,48 %**
	0.66	89,63 %	0 %	**89,63 %**	10,37 %	0 %	**10,37 %**
Profesor.Nombre	0.0	97,78 %	0 %	**97,78 %**	1,48 %	0,74 %	**2,22 %**
	0.33	95,56 %	0 %	**95,56 %**	3,7 %	0,74 %	**4,44 %**
	0.66	78,52 %	0 %	**78,52 %**	21,48 %	0 %	**21,48 %**
Trabajo.Escalafon	0.0	99,26 %	0 %	**99,26 %**	0,74 %	0 %	**0,74 %**
	0.33	97,78 %	0 %	**97,78 %**	2,22 %	0 %	**2,22 %**
	0.66	89,63 %	0 %	**89,63 %**	10,37 %	0 %	**10,37 %**
Trabajo.Nombre	0.0	97,78 %	0 %	**97,78 %**	1,48 %	0,74 %	**2,22 %**
	0.33	94,81 %	0 %	**94,81 %**	4,44 %	0,74 %	**5,19 %**
	0.66	74,07 %	0 %	**74,07 %**	25,93 %	0 %	**25,93 %**

Prob. Campo Faltante es la probabilidad de que no se tenga el valor uno de los campos que se utilizan para hacer extracción focalizada.