Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional

Departamento de Computación

Tesis:

Descubrimiento de relaciones semánticas entre autores y temas de investigación en artículos científicos

Que para obtener el grado de Maestro en Ciencias de la Computación presenta:
Karla del Carmen Durán Arévalo

Director:
Dr. José Guadalupe Rodríguez García

Índice general

Resumen VII

Abstract IX

Agradecimientos XI

1. Introducción 1

1.1. Antecedentes 1

1.2. Planteamiento del problema 3

1.3. Motivación 5

1.4. Objetivos del trabajo 5

1.5. Organización del documento 6

2. Marco teórico 9

2.1. Estado del arte 10

2.2. Procesamiento de lenguaje natural 16

2.3. Extracción de información 18

2.4. Arquitectura general para procesamiento de texto 20

2.5. Stanford parser 24

2.6. Ontologías 25

3. Propuesta de solución **29**

3.1. Extracción de datos 29

3.1.1. Reglas para el anotado de documentos 30

3.1.2. GATE embebido 34

3.1.3. Extracción de anotaciones 35

3.2. Almacenamiento de información 36

3.2.1. Diseño de la base de datos 37

3.2.2. Diseño de la ontología 38

3.3. Relaciones semánticas 41

3.3.1. Comparación de oraciones 42

3.3.2. Poblado automático de la ontología 48

4. Implementación **59**

4.1. Reglas en Jape 60

4.2. Uso de GATE con Java 61

4.2.1. Procesamiento de etiquetas de GATE 64

4.3. Almacenamiento de datos 66

4.3.1. Conexión a la base de datos 66

4.3.2. Manejo de la ontología 66

4.4. Implementación del método para comparación de oraciones 68

4.5. Poblado de la ontología 71

4.6. Interfaz de usuario 72

5. Resultados **77**

5.1. Extracción de datos de artículos científicos 78

5.2. Comparación de oraciones 82

5.3. Funcionamiento del sistema 86

5.3.1. Opciones del sistema 86

5.3.2. Relación entre artículos y autores 87

5.3.3. Análisis de artículos 92

6. Conclusiones y trabajo futuro **97**

6.1. Conclusiones 97

6.2. Trabajo futuro 99

Bibliografía **101**

Resumen

Encontrar relaciones semánticas entre autores de artículos científicos y temas de investigación permite ofrecer servicios de búsqueda de información confiable sobre quiénes son las personas expertas que trabajan sobre un área de investigación y sobre qué artículos científicos ofrecen la información más adecuada sobre un tema. El problema a resolver para ofrecer estos servicios es encontrar los datos apropiados que generalmente son obtenidos de trabajos científicos publicados, que permitan descubrir las relaciones semánticas que formen redes de investigadores sobre distintos temas científicos.

El propósito de esta tesis es crear una herramienta para extraer datos de artículos científicos que permitan encontrar relaciones semánticas entre autores de artículos científicos y sus temas de investigación, lo que permitirá al usuario realizar búsquedas eficientes de artículos científicos y de personas dedicadas a una misma área de investigación.

Se busca obtener únicamente los datos más relevantes que dan información general del artículo, que son título, autor, palabras clave y resumen. Con los datos obtenidos, se realiza un análisis sintáctico de las oraciones del resumen que involucren las palabras clave del documento para formar relaciones semánticas entre los artículos científicos y sus autores. El análisis sintáctico de las oraciones se realiza mediante un nuevo método que obtiene la similitud entre un par de oraciones.

Después de extraer los datos del artículo, analizarlos y encontrar relaciones semánticas, la información generada es almacenada en una ontología con la cual es posible realizar consultas para obtener la información más adecuada para el usuario.

Abstract

Finding semantic relations between authors of scientific articles and research topics provides can provide a reliable information about who are the experts working in an area of research and which scientific articles can offer the best information on a subject.

The problem is that in order to offer services to find the appropriate information these are usually obtained from published scientific papers by discovering semantic relations form networks of researchers in different scientific topics. The purpose of this thesis is to create a tool for extracting data from scientific articles and finding semantic relationships between authors of scientific articles and their research topics, the aim of this proposal is to allow to users to perform efficient searches of scientific papers and researchers engaged in the same area research.

By obtaining the most relevant information in the article, as title, author, keywords and abstract a syntactic analysis of sentences is performed, this analysis involves the document keywords to form semantic relations between scientific articles and their authors. The syntactic analysis of sentences is performed by a new method which obtains the similarity between a pair of sentences. After the analyze of data and discovering the semantic relations, the information generated is stored in an ontology where which it is possible to search the most relevant information for the users.

Agradecimientos

Agradezco a Dios, a mis padres y amigos por todo el apoyo brindado durante esta estapa de mi vida para lograr este objetivo.

Agradezco al Consejo Nacional De Ciencia Y Tecnología y al Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional por brindarme la oportunidad de cursar un posgrado en su programa de maestrías.

Capítulo 1

Introducción

1.1. Antecedentes

La recuperación de información fue mencionada por primera vez por Calvin N. Moores hacia el año de 1950 como "la búsqueda de información en un conjunto de documentos, efectuada a partir de la especificación de un tema"[1]. El mismo autor, señala que la recuperación de información involucra los aspectos intelectuales para describir la información, la especificación para su búsqueda, y cualquier sistema, técnica o máquina que se utilice para llevar a cabo la operación. Dada la definición anterior la recuperación de información abarca tres problemas principales:

- ¿Cómo definir y organizar la información?,
- ¿Cómo especificar la búsqueda? y
- ¿Qué sistemas y técnicas utilizar para este proceso?

Existen muchas formas de resolver cada uno de los problemas mencionados; en cada sistema de recuperación de información se utilizan diferentes enfoques según sean las necesidades de sus usuarios. Hoy en día una herramienta muy útil son los motores de búsqueda disponibles en Internet, los cuales ofrecen a sus usuarios una gran cantidad de resultados en muy poco tiempo. Sin embargo, para llegar a las tecnologías de búsqueda con las que se cuenta actualmente, debieron ocurrir varios acontecimientos importantes, entre los primeros que se pueden mencionar están las máquinas automáticas que se emplearon para analizar el censo de Estados Unidos en 1890 [1]. Un hecho que involucra mayormente las problemáticas de la recuperación de información, aparece a finales de los 40's dónde se obtuvienen documentos del ejercito Alemán durante la segunda guerra mundial y se trata de recuperar y organizar la información científica de tales documentos [2]. La búsqueda

en literatura y la aparición de indexación de citas en publicaciones aparece hacía los años 50's, debido a la creciente brecha científica respecto a la URSS [2]. En el año de 1955, Allen Kent y algunos colegas más presentan las primeras métricas que definen el marco de evaluación de sistemas de recuperación de información. Entre las primeras publicaciones sobre recuperación de información se pueden mencionar las de Cyril W. Cleverdon en 1962 y la de J.W. Sammon sobre una primer propuesta de interfaz visual de un sistema de recuperación de información. Las primeras propuestas de la World Wide Web fueron hechas por Tim Berners Lee en el Consejo Europeo para la Investigación Nuclear. La primera conferencia sobre recuperación de información fue patrocinada por el Departamento de Defensa de los Estados Unidos conjuntamente con el Instituto Nacional de Estándares y Tecnologia en 1992 [2].

El conjunto de datos o documentos sobre el cual se trata de recuperar información es muy importante para que la información que se busca sea confiable. Si el tipo de información que se busca es científica sería sencillo pensar que tal información puede encontrarse en documentos de divulgación científica. Tal como lo señala una carta de 1612, Galileo Galilei fue un pionero en la divulgación científica [2]. En la carta Galileo le explica a su amigo Paolo Gualdo que desea que cualquier persona pueda entender su nueva publicación sobre manchas solares, y por tal motivo él ha escrito la carta en italiano en lugar de latin, idioma comúnmente utilizado en publicaciones científicas. El ejemplo fue seguido más tarde por René Descartes, en su obra 'Discurso del Método"publicada en francés en el año 1637 y por Robert Boyle, en su obra "El químico escéptico"de 1661 la cual fue escrita en inglés.

La publicación científica fue adoptando estándares para mejorar la difusión de los resultados y descubrimientos relevantes científicos. Para hacer una publicación se optó por desarrollar artículos cortos adaptados a un formato estándar que pudieran ser publicados en revistas de circulación mundial escritos en un lenguaje aceptable por toda la comunicación científica [3]. Las partes más importantes de estás publicaciones son título, autores, resumen, palabras principales, introducción, métodos y bibliografía. En la actualidad existen muchas revistas de divulgación científica especializadas en diversos tópicos, y los artículos que se publican en ellas están al alcance de la comunidad científica y otras personas interesadas, sin embargo, la cantidad de información sobre un tópico, puede ser abrumadora si no se tienen las herramientas de búsqueda de información adecuadas. Las herramientas de búsqueda actuales, ofrecen servicios de búsqueda de comunidades científicas y artículos científicos afines a un tópico determinado.

Encontrar comunidades de personas dedicadas a áreas de investigación se ha abordado desde diferentes enfoques, en muchos casos se hace involucrando el análisis del perfil de trabajo del autor y analizando artículos científicos [4, 5]. Además, encontrar los artículos más relevantes sobre un tema no es sencillo, los buscadores comunes disponibles en Internet, regresan resultados muy rápidamente y se espera que sistemas similares sean rápidos o que entreguen mejores resultados, así que se debe acudir a técnicas que permitan realizar búsquedas eficientes [6]. Para resolver este problema, se puede analizar previamente un conjunto de artículos disponibles para un sistema, extraer los datos necesarios, analizar estos datos para encontrar y definir relaciones semánticas y almacenar de forma organizada los datos y relaciones encontradas, que permitan obtener información mediante consultas

al sistema.

1.2. Planteamiento del problema

Extracción de datos

Los artículos científicos de los cuales se pueden obtener datos, son publicados en diversas revistas y conferencias, sin embargo, estas publicaciones tienen formatos distintos, dependiendo de la organización que los publica. Este hecho provoca que estos formatos varíen en el tipo de fuente que utilizan o el orden en que aparecen los datos del artículo tales como título, nombre del autor y sus datos, fecha, editorial, columnas por página, etc. El problema de tener varios formatos es que no existe un patron lingüístico que se aplique a todos los formatos de artículos científicos con el cual se identifiquen los datos que contiene el artículo.

Para que un sistema de extracción de información identifique los datos que se buscan en un documento, se pueden establecer reglas de reconocimiento de patrones lingüístico para que el sistema sepa donde encontrarlos [7]. Sin embargo, para establecer las reglas es necesario que el programador conozca el formato del documento. Otra forma para descubrir los datos es mediante algoritmos de aprendizaje de máquina del tipo supervisados que permitan identificarlos, sin embargo, esta opción se utiliza en documentos que tratan sobre temas específicos, para facilitar el análisis semántico de los datos y la extracción de información [8].

Idealmente se desearía analizar toda la información que contiene un artículo y, de este modo, encontrar la mayor cantidad de datos que permitan establecer relaciones que indiquen mayor información sobre el artículo y sus autores. Sin embargo, analizar tantos datos requiere mucho procesamiento de lenguaje natural, lo que puede ocupar una gran cantidad de tiempo, debido a la complejidad de extraer datos de documentos con diversos formatos y con gran diversidad de contenido. Una alternativa para reducir el tiempo es buscar, en el artículo, los datos que contienen información más significativa, tales como: autor, título, resumen y palabras clave.

Para extraer los datos de los artículos, se utiliza la herramienta de software GATE [9] para implementar métodos que procesen un conjunto de artículos científicos para extraer los datos requeridos. Los métodos permiten la extracción de datos de documentos con los formatos de IEEE, ACM y Springer de artículos científicos.

Encontrar y definir relaciones semánticas

Una vez que se obtienen los datos de un conjunto de artículos, se deben encontrar y definir las relaciones semánticas que permitan obtener información sobre las relaciones que tiene un autor con otro y sus temas de investigación. Los problemas que pueden aparecer para encontrar estas

relaciones radican en la forma de delimitar la comunidad de autores que realmente pertenecen a una misma área de investigación. Uno de los enfoques que se encuentran en la literatura para encontrar estas relaciones es mediante el análisis de coautores de artículos científicos, es decir, a partir de los autores de un artículo se forma una red de investigadores relacionados por ser coautores de artículos, donde lo más complicado es asegurarse que efectivamente tales investigadores pertenecen a la misma área de investigación [4], debido a que una persona que publica un artículo puede cambiar de área con el paso del tiempo y, a pesar de que el autor de un artículo trabaje en un área de investigación durante mucho tiempo, no significa que no pueda relacionarse con otras áreas; además es muy probable que el mismo autor publique artículos acerca de otros temas o bien que su trabajo sea aplicado a otras áreas.

Con toda la información que tiene un artículo, es probable establecer algún tipo de relación entre el tema del artículo y el autor o autores del mismo y, de esta forma, identificar qué autores trabajan sobre los mismos temas de investigación. Sin embargo, si los temas son muy variados, clasificarlos es una tarea compleja [10], probablemente un par de temas sean similares y sea posible relacionarlos entre sí, pero entonces surge la pregunta ¿qué tan similares son dos temas? y responder esta pregunta no es una tarea fácil. Además, el usuario podría buscar cualquier tema y, si el sistema no lo maneja dentro de su conjunto de clasificaciones, el sistema no le ofrecería ningún resultado al usuario, debido a que es difícil interpretar lo que el usuario indica para que el sistema busque información en su base de datos. En cambio, si se mantienen relaciones semánticas entre la información, se pueden obtener mejores resultados para ofrecer al usuario [11].

El enfoque que se propone en este trabajo para encontrar y definir relaciones semánticas, es encontrar las palabras principales del documento mediante la extracción de las palabras principales marcadas en el artículo, posteriormente buscar en el resumen del documento las frases que involucran estas palabras y hacer un análisis sintáctico de dichas frases, este análisis deberá dar como resultado un árbol sintáctico, el cual servirá para ser comparado con otros árboles conteniendo la misma palabra principal, generado por el análisis de otro resumen de artículo. Esta comparación debe dar como resultado información sobre la similitud o relación que tiene un documento con otro y así, establecer la relación que hay entre los autores de los artículos analizados. Cabe resaltar que un análisis de coautores también puede hacerse mediante la inferencia de que dos autores que trabajaron con un mismo autor en otro artículo tienen una relación, sin embargo, esta relación será hecha al obtener la similitud de sus artículos publicados, como ya se mencionó. Para obtener las palabras clave del documento también se utilizará el título del documento.

Almacenar relaciones semánticas

Una vez que se analicen los datos, la información encontrada debe almacenarse en alguna parte, ya sea una ontología o una base de datos, donde además se puedan explotar los datos de manera sencilla y generar más información [7]. Debido a que la información con la que se trabaja son relaciones semánticas, una ontología es la mejor opción, ya que a través de la ejecución de razo-

nadores se puede verificar la validez de relaciones semánticas, lo que no sería posible en una base de datos. Otra ventaja frente a éstas es que establecer restricciones en clases y atributos es más sencillo. Las bases de datos también tienen ventajas frente a las ontologías, tales como velocidad de procesamiento y eficiencia en sus búsquedas de información [12], por lo que una base de datos podría utilizarse para almacenar los datos encontrados extraídos de artículos y ofrecer al usuario los resultados finales de su búsqueda, tales como nombres de artículos, nombres de autores, etc.

1.3. Motivación

Dada la creciente cantidad de trabajos de investigación en el mundo, revisarlos podría ser un proceso lento si lo realiza una sola persona. Debido a la cantidad de nuevos artículos científicos que pueden estar disponibles y además, encontrar los artículos más relevantes en algún tema particular es una tarea aun más ardua, ya que es necesario comparar y analizar muchos artículos.

La razón para buscar artículos científicos es que se tratan de publicaciones fehacientes acerca de temas científicos y para una persona que desea investigar sobre un tema en particular, estos artículos son muy útiles y confiables [13]. Una forma de investigar acerca de un tema es recurrir a los buscadores disponibles en Internet, los cuales analizan datos en cualquier parte de la red. Sin embargo, el resultado de este tipo de búsqueda es que la información obtenida puede ser poco útil y no muy confiable [11], ya que no se sabe quién la está proporcionando. Los usuarios también pueden utilizar buscadores en las páginas Web de bibliotecas o revistas, que limitan sus búsquedas a su material disponible de artículos, sin embargo, los resultados pueden ser muy distintos a los deseados.

Otro modo de investigar acerca de un tema científico puede ser consultando a las personas que se dedican a la investigación de los temas que son de nuestro interés; estas personas podrían encontrarse identificando las áreas de investigación en las que trabaja una persona por medio de los artículos científicos que publican. Algunos sistemas de extracción de información pueden obtener datos de documentos en la Web para armar relaciones que formen biografías de personas famosas [14] y, de manera similar, se podrían extraer datos de artículos científicos que formen relaciones entre temas de investigación y autores de artículos; al mismo tiempo, por medio de esta relación, se pueden establecer otras relaciones entre los artículos que permitan desarrollar búsquedas de artículos relacionados con un tema científico.

1.4. Objetivos del trabajo

General

Diseñar un mecanismo automático para la extracción de datos de artículos científicos que permita descubrir y almacenar relaciones semánticas a través de los principales datos para generar

información.

Específicos

1. Implementar una herramienta para el análisis de artículos científicos publicados en los formatos IEEE [15], ACM [16] y Springer [17], que permita obtener datos tales como autor, título, resumen y palabras clave.

2. Diseñar un método para descubrir relaciones semánticas entre áreas de investigación, autores y palabras clave, que permitan saber qué relación tiene un artículo científico con otro, y que permita saber también en qué medida un autor se relaciona con otro, a través de los temas de investigación con los que se involucran.

3. Realizar el análisis sintáctico de las frases del resumen que contengan las palabras principales de un artículo científico.

4. Diseñar e implementar la herramienta que permita analizar nuevos artículos de investigación y que establezca relaciones semánticas con la información encontrada en la ontología.

5. Permitir la explotación de la información a través de un sistema de consultas para encontrar información sobre relaciones entre autores y temas de investigación.

1.5. Organización del documento

Este documento esta organizado en seis capítulos que explican el planteamiento del problema, solución, implementación y resultados de esta tesis. El segundo capitulo presenta el estado del arte y la teoría relacionada al desarrollo de la tesis que son el procesamiento de lenguaje natural, la extracción de información y ontologías, también se presentan algunas herramientas que ayudan al desarrollo de la tesis, las cuales son la arquitectura general para procesamiento de texo (GATE) y el *Stanford Parser*.

En el tercer capítulo se presenta la propuesta de solución de la tesis, que se compone de la extracción de datos mediante reglas para hacer el anotado de documentos utilizando GATE, almacenamiento de información a través del diseño de una base de datos y una ontología y el descubrimiento de relaciones semánticas mediante la comparación de oraciones y el poblado de la ontología. En el cuarto capítulo se presenta la implementación de la propuesta de solución, la cual involucra el desarrollo de reglas en Jape, el uso de GATE embebido y el procesamiento de las etiquetas que se obtienen con GATE, la implementación de la base de datos y la ontología, así como su manipulación por medio del sistema que se propone para analizar documentos y consultar información. Además se presenta la implementación del método para comparar oraciones que ayuda a realizar el poblado de la ontología.

En el quinto capítulo se presentan los resultados que se obtuvieron del desarrollo de la tesis. Hay tres etapas fundamentales del desarrollo de las cuales se pueden obtener resultados parciales, éstas son la extracción de datos de artículos científicos, comparación de oraciones y el funcionamiento general del sistema desarrollado, el cual incluye el análisis de artículos científicos y las relaciones que se descubren entre artículos y autores. Finalmente en el capitulo seis se presentan las conclusiones de la tesis y el posible trabajo futuro que puede hacerse.

Capítulo 2

Marco teórico

En este capítulo se presenta la teoría relevante para entender el procedimiento que se lleva a cabo durante la extracción de información, el procesamiento de datos y el almacenamiento de información que recupera el sistema. Para familiarizarse con los sistemas que realizan tareas similares a la propuesta en este trabajo, primero se presenta el estado del arte de estos sistemas, así como las tareas que se verán involucradas para relacionar artículos científicos y sus autores, es decir la comparación sintáctica de texto. Enseguida se presentan otros aspectos teóricos relevantes.

Los documentos que se procesan están escritos en lenguaje natural, por lo que se utilizan técnicas de procesamiento de lenguaje natural para la extracción de información. El enfoque que se da en esta tesis para resolver los problemas planteados requiere de procesamiento de lenguaje natural en diferentes niveles de profundidad, por lo que es necesario definir claramente los conceptos de procesamiento de lenguaje natural más comunes que son útiles para entender las técnicas que se utilizan en el desarrollo de la tesis.

Las técnicas de procesamiento de lenguaje natural para extraer información varían de acuerdo a la aplicación que se desee dar al sistema, sin embargo, existe una arquitectura típica de estos sistemas que sirve como guía en la formación de un nuevo sistema. Además es importante saber cuáles son las etapas de la extracción de información porque, dependiendo de cada sistema, algunas etapas pueden eliminarse, mientras que otras pueden ser indispensables para el sistema. Escoger las etapas adecuadas para el sistema que se presenta en esta tesis es importante para mejorar el desempeño del sistema.

La extracción de información generalmente se puede realizar usando módulos bien definidos que se han utilizado ampliamente en trabajos anteriores. Existen herramientas de extracción de información que integran dichos módulos y que ayudan a sus usuarios a formar nuevos sistemas de extracción de información. Entre estas herramientas se puede mencionar GATE, que es muy útil en la obtención de datos de artículos científicos.

Para obtener relaciones entre artículos y autores se procesan los resúmenes de cada artículo; el análisis de cada resumen se hace mediante árboles sintácticos de las frases del mismo y, para obtener estos árboles se utiliza el *Stanford Parser* [18]. Es importante que se entiendan las capacidades del parser y el tipo de árboles que da como resultado, ya que de las características de cada árbol se obtienen métricas de similitud en los textos.

Para almacenar los datos y obtener mejores relaciones entre artículos y autores se utilizan ontologías que permiten almacenar adecuadamente la información, así como enriquecerla. Las ontologías son similares a las bases de datos, pero es necesario definir bien algunos términos que permitan diferenciar entre una y otra para entender las capacidades de cada una.

2.1. Estado del arte

Extracción de datos y ontologías

La obtención de información a partir de un documento PDF (*Portable Document Format*) y su almacenamiento en una ontología, es una tarea que ha sido utilizada en proyectos donde existe información sobre un tema en específico y es necesario obtener datos para luego analizarlos semánticamente y así poder obtener más información automáticamente. Un ejemplo es el sistema gubernamental de Nueva Jersey llamado *OntoStruct* [7] el cual propone el uso de interfaces que faciliten el comercio electrónico, con el objetivo de proveer un método eficiente y efectivo que ofrezca a nuevos negocios las facilidades necesarias para relacionarse con su gobierno estatal.

OntoStruct trabaja a partir de documentos impresos que contienen registros y políticas de regulación, que las personas propietarias de negocios complementan para realizar trámites y consultas. Debido a que en el pasado los documentos han sido utilizados por distintas agencias, se tiene una gran variedad de formatos, tipos y cantidad de información. La arquitectura del sistema se compone de varias etapas. Primero los documentos se transforman a un formato PDF para procesar la información. Una vez procesada se pasa a formatos XML (*eXtensible Markup Language*) para trabajar la extracción de términos, lo cual permite extraer las relaciones entre los términos encontrados y construir automáticamente la ontología para almacenar adecuadamente la información obtenida.

La extracción automática de información no se limita únicamente a documentos PDF, también puede ser hecha a partir de documentos Web. Este es el caso de *Artequakt* [14], cuyo enfoque es utilizado para formar biografías de artistas, reuniendo información de la Web y, con ella, poblar una ontología para posteriormente mostrar al usuario un documento reorganizado con la información encontrada. Los párrafos u oraciones de documentos que utiliza Artequakt pueden ser escogidos manualmente o mediante tecnología apropiada de motores de búsqueda; después se pasan los fragmentos de información a la ontología con metadatos derivados del vocabulario de la ontología, consolidando así la información de la biografía generada. Finalmente, cuando el usuario hace una

consulta al sistema, se generan narrativas de la biografía a través de una interfaz Web.

Redes de autores y artículos de investigación

Un trabajo que se relaciona más con nuestra propuesta es *Ontocopi* [4] que es una herramienta para identificar comunidades con intereses similares por medio del análisis de ontologías de ámbitos de trabajo relevantes. Ontocopi, tiene como tarea identificar una relación entre el autor a y el autor b, donde a pesar de que no hayan trabajado juntos en algún artículo, ellos tienen una relación si ambos han trabajado con el autor c.

Ontocopi utiliza las relaciones semánticas en una ontología para descubrir nuevas conexiones entre objetos. La complejidad de obtener las relaciones puede aumentar en gran medida, debido a que al descubrir una nueva relación y continuar con la búsqueda, nuevas relaciones pueden formarse, lo cual no es malo, sin embargo, ir demasiado lejos desde una persona inicial podría generar relaciones que no tienen nada que ver con la búsqueda principal, debido a que se está infiriendo una relación por medio de nombres de autores y publicaciones, pero no a partir de un análisis semántico del contenido de cada uno de los artículos de investigación que los autores publican. Además, un mismo autor puede trabajar en un mismo tema de investigación durante mucho tiempo, pero en determinado momento cambiar de tema, por lo que relacionar personas sin tomar en cuenta la fecha del trabajo que forma la relación sería incorrecto.

Una forma de mejorar la búsqueda de relaciones puede ser pidiéndole al usuario que introduzca más datos sobre el tipo de relaciones que desea encontrar. La selección de parámetros puede incluso omitir a ciertas personas; éste es uno de los modos de ejecución del sistema; otros modos son el semiautomático y el totalmente automático que son menos efectivos que el manual, pero más fáciles de utilizar.

Otro enfoque propuesto por Ahmedi [19] para modelar redes de coautores extiende la ontología FOAF (*Friend of a Friend*) [20] e implementa reglas SWRL (*Semantic Web Rule Language*) para construir redes de coautores, mediante métricas como exclusividad, frecuencia o pesos de acuerdo al número de ocasiones con las que un autor trabaja con otro. Al aplicar análisis de redes sociales a redes de coautores, se encuentran interesantes datos sobre publicaciones y relaciones entre ellos. Un punto importante de este enfoque es que se toma en cuenta el hecho de que la relación de un autor con otro cambia dinámicamente con el tiempo, así que la magnitud del peso que se le asigna a la relación entre dos autores puede ser modificada mediante nuevas publicaciones registradas en su sistema. Para calcular los pesos que se proponen, se utilizan mecanismos como razonamiento con reglas en Web semántica. Las métricas que se proponen en este enfoque son establecidas mediante ecuaciones, que son aplicadas a las reglas que utilizan los mecanismos de razonamiento en la ontología; entre las métricas más importantes están:

- Exclusividad por publicación: representa el grado con el cual el autor a y el autor b han

trabajado juntos exclusivamente para una publicación particular, es decir que se toma en cuenta que no hay el mismo grado de relación entre dos autores, en un artículo donde trabajan únicamente dos autores que en uno donde trabajan tres o cuatro autores juntos.

- Frecuencia de relación como coautor: se refiere al número de veces que un par de autores han trabajado juntos.
- Frecuencia total de relación como coautor: consiste en sumar la frecuencia de valores con los que un autor ha trabajado con otros autores.

Algunas ontologías ya han tratado de modelar la información que se genera a partir de artículos de investigación, entre ellas se encuentra la ontología SWRC [21]. El objetivo de esta ontología es únicamente servir para que otros trabajos puedan utilizarla e incluso extenderla, de manera que desde su creación ha sufrido varios cambios, tanto en el diseño como en el lenguaje que se utiliza para su implementación. Actualmente OWL (*Ontology Web Language*) se utiliza en una de las implementaciones de esta ontología, ya que es el lenguaje estándar adoptado por W3C (*World Wide Web Consortium*). La ontología SWRC cuenta con seis clases:

- Proyecto: describe el trabajo sobre un tema.
- Tópico: describe un tema de investigación.
- Evento: puede referirse a una conferencia o reunión similar.
- Organización: es el nombre de la asociación dónde trabaja una persona o se lleva a cabo un proyecto.
- Persona: describe a los autores, editores y estudiantes.
- Publicación: describe las características del artículo de investigación.

La ontología cuenta con 42 propiedades de objeto, que además están enriquecidas con una gran cantidad de información adicional, para darle mayor coherencia a la información. Esta ontología y su estabilidad ha sido probada en varios proyectos, entre los que se pueden mencionar: AIFB [22] que la utiliza para organizar personal, publicaciones, proyectos y sus correspondientes relaciones; Ontoware [23], que extiende la ontología SWRC para organizar sus proyectos y desarrolladores de software; y OntoWeb [24], que facilita el intercambio de conocimiento y comercio electrónico.

Otro sistema que se puede mencionar es *Arnetminer* [25], el cual proporciona a sus usuarios servicios de minería de datos en búsqueda de redes sociales de investigadores. El sistema se enfoca en crear un perfil basado en semántica para cada investigador mediante la extracción de información en la Web, integrar datos académicos de múltiples fuentes así como analizar y descubrir información relevante relacionada con cada investigador. La arquitectura de Arnetminer se compone de cinco niveles:

1. Extracción: se enfoca en extraer automaticamente de la Web perfiles de investigadores.
2. Integración: los perfiles extraídos son integrados a trabajos publicados usando el nombre del investigador como identificador.
3. Almacenamiento y acceso: proporciona almacenamiento e indexación para extraer e integrar datos.
4. Modelado: por medio de un modelo generativo probabilístico se obtienen diferentes tipos de información.
5. Servicios de búsqueda: basado en los resultados del modelado de datos, se proporcionan los siguientes servicios.
   - Búsqueda de perfil: muestra información de un investigador tal como datos de contacto, temas de interés para el investigador, historia de su educación, etc.
   - Búsqueda de expertos: indica el conjunto de investigadores expertos en un área específica.
   - Análisis de conferencia: el resultado muestra quiénes son los investigadores que generalmente participan en una conferencia específica, así como los trabajos presentados en ella.
   - Búsqueda de cursos: sugiere al usuario los cursos relacionados con un área especifica.
   - Explorador de tópico: muestra el conjunto de trabajos publicados relacionados con un tema específico.

Para definir los perfiles de investigadores, Arnetminer extiende la ontología FOAF [20] y, para extraer la información de la Web, primero se seleccionan las páginas Web más adecuadas y después se procesan los datos. El procesamiento consiste en separar el texto en palabras y asignar posibles etiquetas descriptivas a cada palabra y, para identificar un tipo de palabra, se utilizan expresiones regulares.

Para modelar los datos que utiliza el servicio de explorador de tópico y búsqueda de expertos, se utiliza el modelo ACT (*Author-Conference-Topic*) [5] el cual tiene tres diferentes estrategias para su implementación. En el primer modelo (ACT1), la información de una conferencia es vista como una estampa asociada con cada palabra de un documento, por lo cual cada autor es asociado con una distribución multinomial sobre tópicos y las palabras en el documento, entonces la estampa de conferencia es generada a partir de un muestreo de tópico; el principio que sigue este modelo es que los coautores de un documento determinan los tópicos de su trabajo y cada tópico genera cierto tipo de palabras, todo esto determina una proporción de las publicaciones que tienen lugar en una conferencia en particular. El segundo modelo (ACT2) se deriva de que cuando un documento se escribe, los coautores usualmente primero determinan el lugar dónde desean publicar su trabajo y entonces escriben su trabajo basado en los temas de la conferencia y los intereses de los autores; cada par de conferencia y autor es asociado con una distribución multinomial sobre tópicos y finalmente un conjunto de palabras es generado desde el muestreo de un tópico. En el tercer modelo (ACT3),

cada autor es asociado con una distribución de tópicos y la estampa de conferencia es generada después de que los tópicos han pasado por un muestreo de todas las palabras en un documento; este modelo está basado en la suposición de que los autores primero escriben un documento y luego determinan dónde publicar su trabajo de acuerdo a los tópicos que el autor aborda.

La tarea de extraer información de artículos científicos mediante GATE ya se ha trabajado anteriormente en el ámbito académico [26]. Sin embargo, únicamente se han desarrollado los métodos para trabajar con un sólo formato de documento y no se realiza ningún análisis semántico de la información extraída, como podría ser la clasificación del artículo en un área de investigación o relaciones entre los autores de los artículos analizados. Para almacenar los datos encontrados, se utiliza una ontología que consta de las clases artículo y autor, así como una base de datos que almacena nombres de autores, sus correos, título del artículo o documento, fecha, etc.

Comparación sintáctica de textos

Los métodos que existen para encontrar la relación que existe entre autores de artículos científicos y sus temas de investigación se han abordado desde varias perspectivas. Un enfoque distinto, a los que se han mencionado, es mediante el análisis de similitud de los artículos científicos. Para encontrar la similitud entre textos se pueden utilizar diferentes técnicas, por ejemplo de cada texto se obtienen algunas características, como palabras más frecuentes o en un análisis más profundo se obtienen los tópicos que se tratan en el texto; posteriormente se pueden comparar estas características y así obtener una relación con otro texto [5, 10].

Otro enfoque como el presentado en DLSITE-2 [27] aplica técnicas de análisis sintáctico para calcular la similitud semántica entre un texto y frases que hipotéticamente podrían encontrarse en el texto. Sistemas similares intentan hacer coincidir árboles sintácticos para decidir la medida de similitud entre oraciones o al menos se apoyan de este tipo de técnicas. DLSITE-2 construye un árbol sintáctico a partir de las frases de un texto y selecciona aquellas que contengan información acerca de conjunto de palabras que pertenezcan a categorías gramaticales que son más relevantes, como verbos, números, adjetivos, etc. De este modo, se eliminan los datos menos importantes y así reducir tiempo de procesamiento.

En esta propuesta se define que un primer árbol sintáctico de una frase está contenido en un segundo árbol sintáctico de una segunda frase si todos los nodos y ramas del primer árbol están presentes en el segundo árbol. Así, el siguiente paso del sistema es determinar si la frase hipotética que se representa mediante un árbol sintáctico, realmente se encuentra en el texto que se analiza. Para este propósito se comparan las raíces de ambos árboles, si ellos coinciden, se procede a comparar sus respectivos nodos hijos, los cuales son segmentos que tienen algún orden de dependencia con su respectivo segmento de raíz.

Con el fin de hacer más flexible el sistema, no se requiere que los pares de segmentos sean explícitamente los mismos, pero si es necesario que estos segmentos no sobrepasen un umbral mínimo

de similitud entre ellos. Este umbral es calculado mediante la medida de Wu-Palmer [27], que utiliza como parámetros el número de nodos de cada camino de cada uno de los árboles comparados. Si el grado de similitud es aceptable, se considera que el significado del segmento de texto analizado es el mismo y se procede a comparar los siguientes nodos hijos del árbol de la hipótesis. Por otra parte si el grado de similitud no es aceptable, se procede a comparar los nodos hijos del árbol del texto con los nodos del árbol de la frase hipotética que ya fueron analizados. El proceso continua hasta que todos los nodos de cada árbol han sido procesados.

Este no es el único proceso de medición que se realiza, también se buscan los pares de nodos en cada árbol que son totalmente idénticos, sin importar si no se encuentran en la misma posición de cada frase. Finalmente, se asignan pesos a los segmentos de las frases, los cuales son calculados empíricamente, asignando menor importancia a los nodos de mayor profundidad localizados en el árbol. En el peso asignado también está involucrada la relevancia de la categoría gramatical de las palabras en la frase, por ejemplo se asigna un mayor peso a los verbos que a un sustantivo. Los pesos para cada categoría gramatical son asignados empíricamente. Cuando el proceso es completado se asigna el grado de similitud entre las frases.

Por otra parte, Aron Culotta y Jeffrey Sorensen [28] proponen un método para detectar y clasificar, entidades y relaciones que hagan una conexión entre sus entidades. Cada relación es instanciada como un árbol de dependencias. Un árbol de dependencias representa las dependencias gramaticales en una sentencia y el método agrega características adicionales a cada nodo del árbol. La razón por la que se utilizan estas representaciones es que ellos suponen que las instancias contienen relaciones similares que compartirán estructuras similares en sus árboles de dependencias; la tarea de la función de núcleo es encontrar la similitud.

Es común referirse a un núcleo de bolsa de palabras, simplemente como un núcleo. Cuando una instancia es más estructurada, como en el caso de los árboles de dependencias, el núcleo se vuelve más complejo; por ejemplo, en una convolución de núcleos, se encuentra la similitud entre dos estructuras por medio de la suma de similitudes de sus sub-estructuras. De modo que, para determinar la similitud entre dos frases, se cuenta el número de secuencias comunes en las dos frases, así como el peso de estas coincidencias por su longitud.

Para detectar y clasificar las relaciones entre entidades, proponen que para cada par de entidades se cree un árbol de dependencia aumentado. Un conjunto de reglas mapean el árbol analizado con el árbol de dependencias, por ejemplo los sujetos son dependientes del verbo y los adjetivos son dependientes del sustantivo. Para generar el árbol de dependencias de cada sentencia se utiliza un analizador de entropía estadístico. Posteriormente, se utiliza una función para determinar la similitud simétrica de los árboles.

Un método basado en árboles para reconocer vínculos textuales, mediante la similitud de sentencias, es propuesto por Rui Wang y Günter Neumann [29]; los vínculos textuales son introducidos como una relación entre expresiones de texto, capturando el hecho de que el significado de una expresión puede ser inferido desde otra expresión. El método trata de encontrar las conexiones po-

tenciales entre relaciones pertenecientes a diferentes capas lingüísticas para diferentes aplicaciones, además está basado en la idea de que en un texto hipotético es más corto que un texto común; no toda la información en el texto es relevante para decidir el vínculo entre un texto y una frase y la diferencia de relaciones, entre los mismos tópicos del texto y la frase hipotética, son de gran importancia.

Este método representa las diferencias estructurales entre el texto y la frase hipotética, eliminando los datos del texto que son irrelevantes para la frase. Una función de similitud triple es aplicada a las relaciones de dependencias entre el texto y la frase, además se calcula la proporción de superposición entre la frase y el texto. El reconocimiento de vínculos generalmente es abordado por medio de dependencia de árboles, tripletas, bolsa de palabras, nivel de palabras, inferencias o análisis de discurso.

En el método que propone Wang y Neumann se aplica una serie de etapas que extraen las dependencias de árboles entre un texto y una frase. La primera estrategia que se utiliza es un método de bolsa de palabras, la segunda estrategia se basa en un conjunto triple de representación de sentencias que expresa la relación de dependencia local encontrada. Una función de similitud obtiene cuántas tripletas de la frase están contenidas en el texto y se basa en la teoría de que cuanto mayor sea el número de coincidencias de elementos de tripletas, más similares son los conjuntos y es más probable que el texto implique a la frase; además, se le da importancia a las diferencias particulares entre el texto y la frase. Para extraer el árbol de dependencias, se construye un conjunto de palabras clave que se usan en todos los sustantivos que aparecen en el texto y la frase; entonces, al armar los árboles de dependencia del texto y la frase, se marcan las palabras clave, de modo que un requisito de este método es que el árbol del texto tenga caminos que contengan todas las palabras clave de la frase hipotética. Las etapas que propone el método se forman de una serie de funciones que obtienen la similitud entre los árboles que forman al texto y a la frase.

Igualmente, un área donde se utiliza este tipo de técnicas para encontrar la similitud entre árboles de oraciones es el área de bioinformática, tal como se presenta en RelEx [30], que produce árboles de dependencia y utiliza un pequeño número de reglas para cada árbol, con el fin de extraer relaciones en procesamientos metabólicos o modelos de enfermedades. RelEx crea relaciones candidatas por medio de la extracción de pares de proteínas conectados por ramas de árboles de dependencia, obtenidos de documentos biomédicos. Posteriormente, se utilizan reglas que reflejan las relaciones más usadas en el idioma inglés que describen la información que se busca. Estas relaciones son aquella donde un objeto X activa a un objeto Y, la activación de un objeto X por un objeto Y y la interacción entre un objeto X y un objeto Y.

2.2. Procesamiento de lenguaje natural

A través de la historia de la humanidad el conocimiento ha sido comunicado y almacenado en forma de lenguaje natural. Hoy en día esto no ha cambiado a pesar de guardarse en forma

electrónica o digital, lo cual significa un gran avance debido a que las computadoras pueden ser de gran utilidad para el procesamiento de este conocimiento; por tal razón se dedican muchos esfuerzos al desarrollo de la ciencia que se encarga de habilitar a las computadoras para entender el texto. En función del enfoque práctico versus teórico del grado en el cual se espera lograr la comprensión, esta ciencia es llamada procesamiento de lenguaje natural, procesamiento de texto, tecnologías de lenguaje o lingüística computacional [31].

El entendimiento del lenguaje natural escrito se conoce en el ámbito de la inteligencia artificial como "Procesamiento de Lenguaje Natural"(*PLN*) y se enfoca en la recepción de texto, cuyo contenido es interpretado léxica, sintáctica y semánticamente en función del conocimiento que se tiene del lenguaje, del contexto y de la persona que lo expresa, además del conocimiento ordinario [32].

Entre las principales áreas del procesamiento de lenguaje natural se puede mencionar [31, 33]:

- Recuperación de información.
- Interfaces de lenguaje natural.
- Traducción automática.
- Extracción de información.
- Sistema de preguntas - respuestas.
- Generación automática de resúmenes.

En cuanto a los sistemas de recuperación de información, su efectividad puede ser mermada por algunas desventajas del lenguaje natural, éstas son la variación lingüística, que se refiere a la posibilidad de utilizar diferentes palabras o expresiones para comunicar una misma idea y la ambigüedad lingüística, que se produce cuando una palabra o frase se puede interpretar de diferentes formas [33].

En el estudio del PLN intervienen diversas disciplinas tales como matemáticas, lingüística, ingeniería informática, filosofía y psicología; tanto desde un enfoque computacional como lingüístico, se utilizan técnicas de inteligencia artificial como modelos de representación de conocimiento y razonamiento, lenguajes de programación declarativos, algoritmos de búsqueda y estructuras de datos. Sin embargo, el uso de estas técnicas no aportaría soluciones adecuadas sin una concepción profunda del fenómeno lingüístico, por lo que el estudio del lenguaje natural se estructura normalmente en los siguientes niveles de análisis [34]:

- **Morfológico**: consiste en detectar la relación que se establece entre las unidades mínimas que forman una palabra llamadas morfemas y determina el género, número o flexión que componen las palabras. La flexión de las palabras puede ser nominal o verbal; en la primera

se aplican pronombres, sustantivos y adjetivos; se dice que una palabra se compone de un lexema o raíz y de morfemas gramaticales de género y número. La flexión verbal es también llamada conjugación de los verbos y se compone de una raíz de morfemas gramaticales que indican tipo, tiempo, modo, persona, número y género.

- **Sintáctico**: tiene como función etiquetar cada uno de los componentes sintácticos que aparecen en una oración, para después analizar cómo las palabras se combinan para formar construcciones gramaticalmente correctas. Los componentes sintácticos son formados por unidades léxicas que forman la oración, mientras que la gramática se forma por medio de un conjunto de reglas que para definirse utilizan sintagmas nominales, sintagmas verbales y determinantes; como resultado de aplicar estas reglas se obtiene un árbol sintáctico de la oración.

- **Semántico**: este nivel estudia el significado de una palabra y el de una frase a partir de los significados de cada una de las palabras que componen una oración [33]. Encontrar el significado no es una tarea sencilla, por lo que sería deseable distinguir entre significado independiente y significado dependiente del contexto. La semántica busca identificar el significado de la frase, ignorando la influencia del contexto. Si se desea que exista una simetría entre la estructura semántica y la estructura sintáctica de la oración, a partir del árbol generado por el análisis sintáctico se genera una estructura arbórea con las mismas características sobre la cual se realizará un análisis semántico. En cambio si no se desea tal simetría, en la estructura sintáctica que se generó se aplica una serie de transformaciones con las cuales se genera la representación semántica de la oración. Así, un analizador semántico procesa cada palabra para obtener una representación del significado de la frase.

- **Pragmático o contextual**: su función es añadir información al análisis del significado de la frase en función del contexto donde aparece. Este es uno de los niveles más complejos, ya que incorpora al análisis semántico la aportación significativa que los participantes que utilizan el lenguaje, así como información sobre las relaciones que se dan entre los hechos que forman el contexto y otras entidades.

2.3. Extracción de información

El procesamiento de lenguaje natural es también útil en los sistemas de extracción de información [8], los cuales tienen como función encontrar hechos muy específicos en textos acerca de un dominio de interés, sin la necesidad de que una persona los analice. Esto es útil sobre todo si se desean analizar muchos documentos; hoy en día la red de Internet es muy amplia y existen muchos usuario que publican diariamente documentos nuevos, incluso los periódicos publican sus artículos en Internet; si se deseará saber si en alguna parte del mundo ocurrió un hecho específico muy importante sería necesario leer todos los artículos de los periódicos del mundo, lo cual sería imposible. Con los sistemas de extracción de información, se podrían analizar todos estos documentos y obtener la información que se busca.

En cambio, los sistemas de recuperación de información son usados para obtener los documentos más relevantes sobre un tema específico de modo que el PLN se utiliza para describir el contenido de un documento y representar la consulta formulada por el usuario, de modo que ambas descripciones puedan ser comparadas y así presentar al usuario aquellos documentos que satisfagan mejor su necesidad de información. Se pueden mencionar dos principales aproximaciones utilizadas en el PLN para la recuperación de información, la aproximación estadística y el enfoque lingüístico [33]. El procesamiento estadístico del lenguaje natural [35] representa el modelo clásico de los sistemas de extracción de información y se caracteriza porque cada documento está descrito por un conjunto de palabras clave denominadas términos índice. Por otra parte, el procesamiento lingüístico del lenguaje natural, se basa en la aplicación de diferentes técnicas y reglas que codifican de forma explicita el conocimiento lingüístico [36], de modo que los documentos son analizados en diferentes niveles lingüísticos e incorpora al texto las anotaciones propias de cada nivel.

Una búsqueda sencilla de los documentos más relevantes sobre un tópico a veces no es suficiente y es necesario identificar cierta información en el contenido de un documento recuperado; donde un sistema de extracción de información es útil y su desempeño puede ser mejor si el propio documento contiene información específica sobre el tema.

Las principales tareas de un sistema de extracción de información son [37]:

- Reconocimiento de entidades nombradas: se insertan etiquetas en el texto que marcan cada palabra que representa un nombre propio. Además, estas entidades se clasifican para determinar su tipo, como una organización, nombre de una persona, lugar geográfico o fecha.
- Construcción de plantilla de elementos: extrae la información básica relacionada con la entidad nombrada. Es decir, a cada entidad se le asocia un conjunto de atributos que la describen. La información que se le asocie a cada entidad depende del tipo de información que se busca finalmente; de manera que una plantilla con información sobre una entidad nombrada, como una organización, pueda ser completada.
- Construcción de plantilla de relaciones: extrae la relación que existe entre las entidades nombradas. Con la información obtenida de la etapa anterior, las plantillas con información de cada entidad deben formar una relación, la cual también es determinada por el tipo de información que se busca.
- Construcción de plantilla de escenarios: identifica información específica de un evento, la cual fue obtenida de las plantillas completadas anteriormente que tienen información a cerca de eventos y se relacionan a una entidad nombrada en particular.
- Identificación de correferencia: captura la información en las expresiones de correferencia, todas mencionadas de una entidad dada.

Para construir un sistema de extracción de información [8], existen dos enfoques, el enfoque de ingeniería del conocimiento, donde es necesario analizar un corpus y con él construir gramáticas,

y el enfoque de métodos empíricos, el cual utiliza métodos estadísticos y algoritmos que puedan generar reglas a partir de un corpus anotado manualmente que pueda ser interpretado por un sistema de PLN. La arquitectura típica de estos sistemas puede observarse en la figura 2.1 y es descrita a continuación:

1. Segmentar componentes léxicos y segmentar oraciones: a partir de un texto se obtienen las cadenas de palabras (componentes léxicos) y fronteras de oraciones.
2. Análisis morfológico: determina la forma, clase o categoría gramatical de cada palabra de una oración.
3. Clasificación de sustantivos: determina si una palabra es un nombre propio como un lugar, fecha, nombre de persona, compañía etc.
4. Análisis sintáctico de superficie: las oraciones se segmentan en sintagmas nominales, sintagmas verbales y partículas. los sintagmas nominales identifican sustantivos y pronombres, los sintagmas verbales identifican eventos o acciones y las partículas se refieren a los artículos, preposiciones, conjunciones y adverbios.
5. Reconocimiento de entidades y eventos: del conjunto de sintagmas nominales se reconocen las entidades, como compañías o lugares geográficos, y de los sintagmas verbales se reconocen los eventos.
6. Resolución de correferencia: se identifican las relaciones entre entidades y eventos o acciones.
7. Reagrupación: finalmente se llenan las plantillas previamente diseñadas para asociar entidades con atributos.

2.4. Arquitectura general para procesamiento de texto

Existen herramientas de software dedicadas al procesamiento de lenguaje natural y a la extracción de información; una de ellas es GATE (*General Architecture for Text Engineering*) [9], desarrollada en 1995, está basada en Java y es ampliamente usada por diversas comunidades de científicos, compañías, profesores y estudiantes. GATE es usada para extraer información general de un texto y fue especialmente desarrollada para la extracción de entidades nombradas, por ejemplo título del documento, direcciones, correos electrónicos, etc. Entre otras funciones de GATE se puede mencionar que permite construir, anotar o etiquetar corpus, el cual se define como un conjunto de textos almacenados en formato electrónico y agrupados con el fin de estudiar una lengua o una determinada variedad de lingüística [38].

La arquitectura de GATE está basada en componentes bien definidos [9], es decir módulos de software que sirven como interfaz para desarrollar una gran variedad de herramientas. Estos

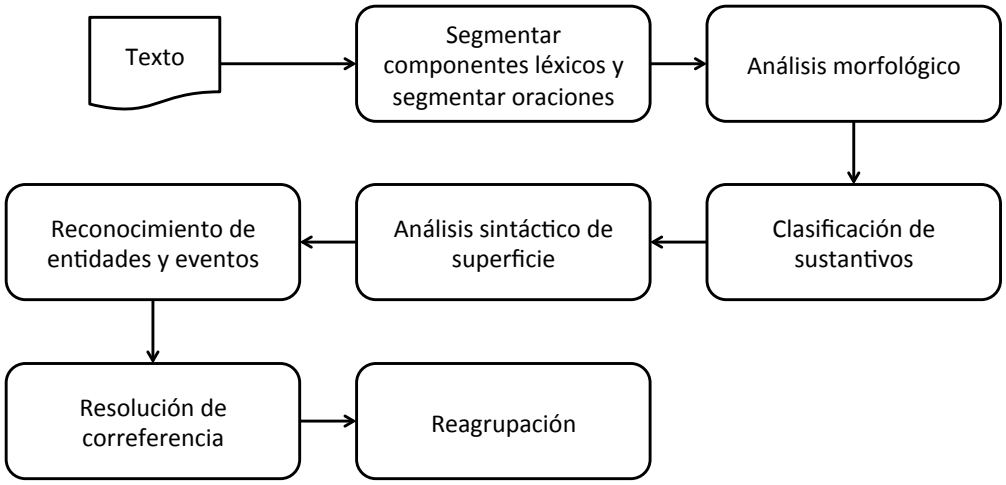

Figura 2.1: Arquitectura típica de sistemas de extracción de información

módulos se clasifican en *recursos de lenguaje*, como lexicones y corpus, *recursos de procesamiento*, como algoritmos o entidades para segmentar oraciones y palabras, y *recursos visuales* usados para interfaces gráficas. Los módulos que conforman la arquitectura de GATE son [39]:

- El administrador de documentos GATE: tiene como función comunicar los componentes de GATE y almacenar los textos o información utilizada o generada en las aplicaciones de PLN implementadas con GATE.
- La interfaz gráfica de GATE: es utilizada por el usuario para construir y ejecutar componentes gráficos en aplicaciones de PLN. Permite mostrar los resultados del procesamiento de corpus, editar parámetros de componentes, integrar nuevos componentes, etc.
- CREOLE (*Collection of Reusable Objects for Language Engeenering*): es un conjunto de herramientas u objetos útiles en el procesamiento de lenguaje natural, que el usuario puede reutilizar para elaborar aplicaciones.

Otros recursos con los que cuenta GATE son aplicaciones y almacenamiento de datos. Las aplicaciones en GATE son conjuntos de recursos de procesamiento agrupados y ejecutados en sucesión sobre algún recurso de lenguaje; estos pueden ser una *tubería* o una *tubería de corpus* [8]. Una aplicación de tipo tubería se ejecuta únicamente sobre un documento, mientras que la tubería de corpus es útil para procesar un conjunto de documentos.

Por otra parte, la información generada de la ejecución de recursos de procesamiento sobre documentos, como lo es la segmentación de un párrafo u oración donde se separan palabras o números, se le llama *anotación* [9]. Otro ejemplo de anotación es la que genera un etiquetado

morfosintáctico, el cual asigna una categoría gramatical a las palabras encontradas en el texto analizado. Una anotación consiste de un ID, un tipo, un nodo inicial, un nodo final y un conjunto de características. El ID se utiliza para identificar la anotación en el documento, el tipo denota la clase de anotación generada, los nodos inicial y final, indican las posiciones inicial y final de la anotación en el documento y las características, que son atributos con sus respectivos valores, proporcionan mayor información sobre la anotación. Las anotaciones pueden ser almacenadas en formatos XML.

El soporte de ontologías también ha sido incorporado a GATE [40]. Las ontologías normalmente son usadas por recursos de lenguaje, sin embargo, son muy diferentes a los documentos y corpus que generalmente utiliza GATE. Se incluye el API para trabajar con ontologías y también una serie de paquetes con implementaciones. Los métodos que se proveen permiten acceder a la jerarquía de clases de la ontología, listando las instancias y propiedades. Las implementaciones de ontologías están disponibles por medio de *plugins*.

ANNIE

ANNIE (*A Nearly-New Information Extraction System*) es el sistema de GATE para extraer información, el cual es un conjunto de módulos CREOLE para hacer extracción. Los módulos que incluye tienen como función [8]:

- Segmentar palabras: separa el texto en segmentos de palabras, números o signos de puntuación; también permite definir segmentos más complejos mediante el uso de JAPE.
- Segmentar oraciones: como su nombre lo indica, permite separar las oraciones de un texto, las cuales son reconocidas mediante signos de puntuación.
- Hacer análisis morfosintáctico: utiliza el conjunto de etiquetas *Penn Treebank*, con el cual se determina qué tipo de categoría gramatical tiene una palabra; cabe mencionar que el idioma que reconoce este conjunto de etiquetas es el inglés.
- Categorizar sustantivos: este módulo funciona mediante listas de palabras previamente clasificas, con las cuales se intenta hacer coincidir la palabra que se intenta categorizar y así encontrar la categoría de la palabra. Los nombres de los archivos que contienen las listas de palabras están en el archivo llamado `list.def` y los renglones en este archivo son de la forma `nombreDelArchivo.lst:categoría:subcategoria`.
- Hacer análisis sintáctico de superficie: para reconocer los sintagmas nominales y verbales de la oración, se utilizan gramáticas JAPE.
- Reconocer entidades y eventos: para reconocer entidades, se utiliza un archivo llamado `main`, el cual contiene la lista de gramáticas JAPE para hacer el reconocimiento.

- Resolver correferencia: en este módulo se trata de identificar equivalencias entre entidades ya reconocidas, determinando atributos y relaciones entre las entidades. Existen diferentes tipos de correferencia, una de las más comunes es la llamada anáfora, la cual sucede cuando una de las entidades es mencionada de forma diferente a cómo se ha mencionado anteriormente, por ejemplo por medio de un pronombre.

JAPE

El lenguaje JAPE (*Java Annotation Patterns Engine*) permite efectuar transformaciones sobre anotaciones en texto a través de expresiones regulares [40]. La gramática de JAPE es un conjunto de fases, compuestas por reglas que realizan una acción al encontrar un patrón establecido, las cuales se ejecutan sucesivamente sobre un texto anotado. La regla se compone de dos partes, en la primera (escrita al lado izquierdo, LHS) se define qué patrón se busca, la segunda (escrita al lado derecho, RHS) expresa la acción a ejecutar cuando el patrón es encontrado. La parte de la regla que especifica el patrón se indica mediante una expresión regular, mientras que en la acción se indica el nombre de la anotación que se agrega al patrón encontrado y los atributos que se desean asignar a la anotación; además, es posible indicar otras acciones mediante código de Java. La figura 2.2 muestra un ejemplo de una regla que obtiene una palabra escrita con todas sus letras en mayúscula seguida de un verbo conjugado en pasado simple.

```
1  Phase: nombreaccion
2  Input: Token
3  Options: control = appelt
4
5  Rule: Nombreaccion
6  (
7  {Token.kind==word, Token.orth!=allCaps}
8  {Token.category!=VBD}
9  ):nombresacciones
10  -->
11 :nombreacciones.Nombres =
12 {kind = "nombreaccion", rule = "Nombreaccionesr"}
```

Figura 2.2: Ejemplo de regla en Jape

A cada una de las reglas se le asigna un nombre de fase, con el cual el sistema identificará la regla. Cada una de las reglas se ejecuta en cascada, de acuerdo al orden en que su nombre de fase aparece en el archivo principal, donde se buscan las reglas que debe ejecutar el módulo de reconocimiento de entidades. Después de dar un nombre a la fase, se indica qué tipo de anotaciones se recibirán como entrada para que la regla procese; en el ejemplo anterior, el tipo de entrada es Token, es decir que la regla procesará únicamente Tokens; sin embargo, se pueden indicar varios datos, incluso anotaciones que el mismo programador haya establecido con reglas diseñadas por él mismo.

Existen varios modos de ejecutar JAPE, entre los cuales están *appelt*, *brill* y *first*. Cuando existen varias reglas que coinciden con un mismo patrón, es necesario indicar qué regla es la que se debe aplicar para dicho patrón, si se desea que sólo una regla sea aplicada; JAPE debe ejecutarse en modo *appelt* y entonces se debe tener una prioridad en las reglas, para que aquella que tenga la mayor prioridad sea la que se aplique. En cambio, si se desea que todas las reglas que coincidan con el patrón que se busca sean aplicadas, el modo de ejecución debe ser *brill*. Otra opción es que la primera regla que coincida con el patrón que se busca sea la que se aplique, este es el modo de ejecución *first*.

Enseguida se escribe la regla, que consta de un nombre seguido de la LHS, una separación indicada por " − − > " y la RHS. Para indicar el patrón que busca la regla, en la LHS se escribe la secuencia de anotaciones que la regla debe identificar; cada anotación o propiedad de una anotación con una valor dado se encierra entre llaves y si se desea que una anotación que se procesa tenga varias características éstas se escriben dentro de la misma llave, pero separadas por comas.

Es posible que se desee indicar que una o más anotaciones aparezcan varias veces una tras otra; para ello, la serie de anotaciones que se repetirá se encierra entre paréntesis y a continuación se coloca el operador que indica la cantidad de veces que la regla aceptará la repetición. El operador que indica que la aparición de las anotaciones es opcional es "?"; para indicar que las anotaciones pueden aparecer una o más veces se utiliza el operador "+"; si en cambio se quiere indicar que puede aparecer cero o más veces se utiliza "∗"; sin embargo, la utilización de estos dos últimos operadores puede ocasionar un ciclo infinito de búsqueda por lo que se recomienda utilizar "$[a, b]$" en lugar de ellos, donde a indica el número mínimo de apariciones y b el número máximo. Otro operador muy útil es el "|", que indica que entre una serie de anotaciones separadas por este operador puede aparecer una u otra. Para indicar los valores de las características de una anotación, están disponibles los operadores de igualdad (" == " y "! = ") y de comparación (" < ", " <= ", " >= ", " > ").

2.5. Stanford parser

Un analizador sintáctico o *parser* de lenguaje natural resuelve la estructura gramatical de una sentencia [41], identificando el sujeto u objeto de una oración, así como los verbos involucrados en la sentencia. Para obtener la estructura de una oración es necesario darle un significado a cada palabra y esta tarea no es sencilla, ya que una palabra puede tener un significado distinto dependiendo de las palabras con las que este relacionada o incluso el contexto de la oración. Asegurar que una palabra corresponde a una sola categoría gramatical es complicado, por lo tanto es común que los analizadores sintácticos entreguen resultados estadísticos sobre la categoría que asignan a cada palabra de una oración.

El *Stanford Parser* [18] es un analizador sintáctico estadístico y provee un API para ser utilizado en Java. Este parser implementa un modelo de producto factorizado [41] para encontrar

dependencias léxicas entre las palabras de la oración, sin embargo, la implementación puede ser utilizada únicamente como un analizador estocástico gramatical, libre de utilizar un contexto más preciso; además, proporciona una estructura de árbol para representar las relaciones gramaticales de la oración.

La figura 2.3 muestra un ejemplo de un árbol generado con el *Stanford Parser*, a partir de la oración:

```
The tiger is a carnivorous predator and is the largest feline in the world
```

Además, el árbol que muestra la figura 2.3 es el mismo tipo de árbol que será utilizado para trabajar en un método para la comparación de oraciones que ayude a encontrar relaciones semánticas entre artículos científicos.

```
(ROOT
  (S
    (NP (DT The) (NN tiger))
    (VP
      (VP (VBZ is)
        (NP (DT a) (JJ carnivorous) (NN predator)))
      (CC and)
      (VP (VBZ is)
        (NP
          (NP (DT the) (JJS largest) (NN feline))
          (PP (IN in)
            (NP (DT the) (NN world)))))))
```

Figura 2.3: Ejemplo de árbol generado con el *Stanford Parser*

El cuadro 2.1 muestra las etiquetas que el *parser* asigna a los nodos del árbol sintáctico, de acuerdo a la parte de la oración que le corresponde [41]. Los creadores del proyecto *Stanford Parser* aplicaron su trabajo en el lenguaje inglés, sin embargo, con el tiempo el *parser* se ha extendido a otros lenguajes, tales como el chino, alemán, árabe, italiano, búlgaro y portugués. El *parser* está disponible bajo la licencia GNU (*General Public License*).

2.6. Ontologías

Una ontología es una especificación formal de una conceptualización [42], la cual se refiere a un modelo abstracto de cómo piensan las personas acerca de un objeto en el mundo, generalmente restringido a un área particular, mientras que una especificación se refiere al concepto y a la relación del modelo abstracto que está dado en términos y definiciones. Las ontologías fueron diseñadas

para compartir y reutilizar bases de conocimiento [43], sin embargo, esto no se ha logrado como se esperaba, ya que los diseñadores de ontologías plasman su propio enfoque del objeto que se modela y, en ocasiones, no es posible reutilizar la misma base de conocimiento [12].

Las ontologías contienen información sobre el tipo de objeto, sus propiedades y las posibles relaciones que pueden existir entre los objetos en un dominio particular; también pueden ser la representación de vocabulario [43]. Para modelar correctamente la representación de conocimiento, se debe analizar muy bien el dominio en el que se trabaja y así construir un lenguaje de representación de conocimiento basado en análisis, para lo cual es necesario asociar términos con conceptos y relaciones que serán organizados en una ontología, de forma que sea posible distinguir una sintaxis para codificar conocimiento en términos de dichos conceptos y relaciones [44].

Por otra parte, para construir una ontología existen varios métodos de análisis que permitirán encontrar las entidades, propiedades de objetos y sus relaciones, uno de ellos es el *análisis relacional*, el cual descubre y agrupa las principales relaciones, tales como sinónimos o hipónimos y entonces construye una ontología por medio de otros métodos, por ejemplo el *agrupamiento*, el cual reúne las palabras principales de un documento en grupos y selecciona los conceptos más representativos de cada grupo. Otro método es el *análisis formal de conceptos* que usa una matriz de relaciones binarias entre recursos de información y vocabulario para generar un conjunto de conceptos supremos, que organizados jerárquicamente forman una ontología [45].

Un área importante donde las ontologías son utilizadas es en el entendimiento de lenguaje natural [44]. Al utilizar lenguaje natural, existe una gran variedad de formas para expresar una idea, incluso en diferentes idiomas. Un buen diseño de una ontología, que modela el dominio de conocimiento del lenguaje, ayuda a eliminar la ambigüedad en la representación del conocimiento [11].

Etiqueta	**Parte de la oración**
Root	Raíz de la oración
S	Sentencia
VP	Sintagma verbal
NP	Sintagma nominal
CC	Conjunción copulativa
CD	Número cardinal
DT	Determinante
EX	Existencia *there*
FW	Palabra extranjera
IN	Preposición o conjunción subordinante
JJ	Adjetivo
JJR	Adjetivo, comparativo
JJS	Adjetivo, superlativo
LS	Marcador de lista
MD	Modal
NN	Sustantivo, singular
NNS	Sustantivo, plural
NNP	Nombre propio, singular
NNPS	Nombre propio, plural
PDT	Predeterminante
POS	Terminación posesiva
PRP	Pronombre personal
PRP$	Pronombre posesivo
RB	Adverbio
RBR	Adverbio, comparativo
RBS	Adverbio, superlativo
RP	Partícula
SYM	Símbolo
TO	*to*
UH	Interjección
VB	Verbo, forma simple
VBD	Verbo, tiempo pasado
VBG	Verbo, gerundio o presente participio
VBN	Verbo, pasado participio
VBP	Verbo, presente singular diferente de tercera persona
VBZ	Verbo, presente singular en tercera persona

Cuadro 2.1: Etiquetas que son asignadas a los nodos del árbol sintáctico por el *Stanford Parser*, de acuerdo a la parte de la oración que le corresponde.

Capítulo 3

Propuesta de solución

En este capítulo se presenta el diseño de un sistema que sea capaz de extraer datos de documentos científicos, almacenar esos datos en una base de datos y desarrollar métodos para poblar automáticamente una ontología diseñada para almacenar las relaciones semánticas entre autores y artículos cientificos descubiertas por el propio sistema. Además, mediante consultas a la ontología debe ser posible encontrar nuevas relaciones semánticas.

La figura 3.1 muestra el diagrama de flujo general para el procesamiento de documentos, inicia con la introducción de uno o más artículos científicos. El proceso continua con la extracción de datos de los documentos, la cual consiste del anotado de los documentos mediante GATE y la extracción de anotaciones de documentos. Los datos extraídos se almacenan y también se utilizan para hacer poblado automático de la ontología. La extracción de datos, el diseño de la base de datos y la ontología, así como los métodos para hace poblado de la ontología se describen en las siguientes secciones.

3.1. Extracción de datos

La extracción de datos es una parte fundamental para obtener relaciones entre autores y artículos científicos. En esta sección se describe la metodología que se sigue para obtener los datos requeridos de artículos científicos, el tipo de artículos con que se trabaja, y las razones para seguir esta metodología. La extracción de datos se limita a un conjunto de artículos con ciertas características, debido a que los diferentes estilos de escritura en que es posible presentar un trabajo escrito hacen complicado establecer reglas generales que sirvan para identificar todos los tipos de series de palabras que forman el nombre de un autor, el título del trabajo, el resumen y las palabras clave.

Para identificar los datos del artículo se utiliza GATE embebido implementado en Java, poste-

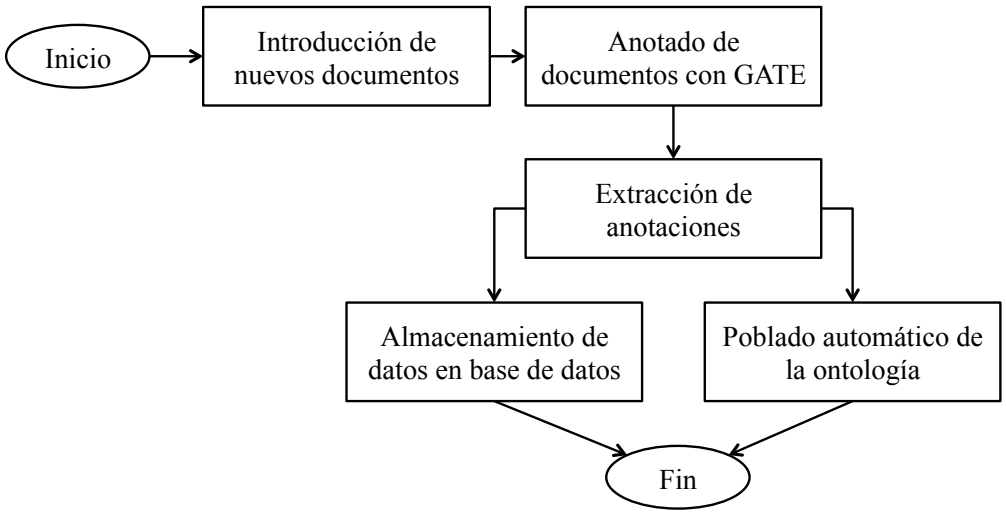

Figura 3.1: Diagrama general del procesamiento de documentos.

riormente los datos extraídos se procesarán extendiendo la aplicación en Java. La parte fundamental es el desarrollo de las reglas escritas en JAPE que encuentran los patrones para identificar los datos del documento. Una vez que se hace el anotado de documentos, se obtienen los datos de cada documento en la aplicación de Java.

En esta sección se presentan las etapas para obtener los datos de los documentos, en la subsección nombrada reglas de extracción de datos, se explican cuáles son las reglas que se deben implementar para anotar los documentos. En la subsección nombrada GATE embebido se hace una breve explicación de cómo utilizar GATE en Java y algunas consideraciones que explican qué módulos de GATE se utilizan para procesar los artículos. En la subsección nombrada extracción de anotaciones con Java se mencionan algunas consideraciones que se toman en cuenta para finalmente poder extraer los datos reales de cada documento.

3.1.1. Reglas para el anotado de documentos

Como se ha venido mencionando, los datos que se requiere extraer de cada documento son título, autores, resumen y palabras clave, sin embargo, para obtenerlos es necesario considerar otras reglas que identificaran otros tipos de datos para después obtener correctamente los datos que originalmente se desean. Estos datos son correo electrónico, universidad y sección. Un correo electrónico se refiere a una etiqueta que identifica el correo electrónico de un autor, ésta no trata de identificar todos los formatos en los que se puede escribir un conjunto o serie de correos, únicamente sirve como dato auxiliar para identificar un renglón con una serie de autores, lo mismo sucede con el dato Universidad, no trata de obtener únicamente universidades, sino cualquier organización a la que pertenezca un autor y que pueda formar parte de la oración que contiene el nombre del autor.

Por otra parte la Sección, es una anotación que es útil para encontrar dónde finalizan las palabras clave y para extraer el resumen.

Los documentos que se procesan son artículos publicados en IEEE, ACM y Springer, disponibles en PDF. Cada uno de estos tipos de artículos siguen diferentes estilos para presentar los datos que se desean obtener, sin embargo, no hay alguna característica que permita conocer rápidamente donde se ha publicado el documento que se analiza. Además los estilos de escritura no son exclusivos de cada publicación y generar una regla para cada estilo podría derivar en procesamiento inútil del documento, ya que la diferencia entre cada regla sería muy poca. Por tal motivo, a pesar de manejar documentos de distintas publicaciones, se aplica una misma regla en la búsqueda de los datos.

Correo electrónico

Una dirección de correo electrónico se compone de un nombre con el cual se identifica al usuario, un símbolo de arroba y el dominio al que pertenece la dirección. El nombre del usuario tiene muchas variaciones, sin embargo, en etapas anteriores a aplicar la regla que se propone, el correo electrónico se segmenta en palabras, números, espacios, etc. De modo que a pesar de ser un conjunto de caracteres unidos, la segmentación permite procesar el correo como una serie de palabras, números, signos de puntuación o cadenas de caracteres. El patrón que identifica el nombre de usuario entonces se compone de una serie de palabras, números y otros caracteres, el símbolo de arroba se puede considerar como una cadena de caracteres y el dominio, nuevamente una serie de palabras, números, y signos de puntuación.

Universidad

Una organización a la cual pertenece un autor se identifica principalmente por contener las palabras “University”, “Institute”, “Department”, “Faculty”, “Academy”, “Center”, “Research”, “Studies”, etc. Las palabras que se incluyen para encontrar la organización, podrían aumentar si se tomara en cuanta un conjunto más amplio de documentos para analizarse. Cabe resaltar que estas palabras pueden aparecer en cualquier lugar de la oración, ya que hay que recordar que la regla busca una serie de palabras, por lo tanto es necesario crear la regla que ubique la palabra al inicio de la oración y otra regla que ubique la palabra al final de la oración. Otros detalles de la regla es que las palabras que contiene el nombre completo de la universidad u organización son sustantivos cuya primera letra es mayúscula, a excepción de palabras comunes como “of”, que puede iniciar en minúscula, también puede contener comas y espacios.

Sección

Una sección en el documento, es la división entre una parte del documento y otra, la finalidad de crear esta anotación es que en algunas ocasiones identificar donde termina una serie de palabras clave es complicado, debido a que no todas éstas terminan con un punto u otro signo. En el caso del resumen sucede lo mismo, y es que su contenido en tan variado que fácilmente puede confundirse con otra parte cualquiera del documento.

La aplicación que se le da a esta anotación permite también conocer el tipo de secciones que se desean etiquetar, es decir, no es necesario etiquetar cada división o título del documento para marcar una nueva sección, ya que sólo es útil la primera después del resumen o palabras clave. Esta sección generalmente es "Introduction", en otros formatos es el "Background", y si las palabras clave se encuentran al final del documento entonces pueden estar seguidas de la sección "References". Es útil notar en esta anotación que se trata de títulos y por lo tanto no están seguidas de nada más que el siguiente renglón. Otro caso que también marca el final de una serie de palabras clave es una línea en el documento, la cual es decodificada como un símbolo, quizá desconocido. También es probable que antes del nombre de la sección se presente el número de la sección.

Título

Un título generalmente se encuentra al principio del documento, esto es una ventaja porque un título puede contener una gran variedad de palabras como sustantivos o verbos, en mayúsculas o minúsculas, y pueden componerse de varios renglones, sin embargo, una de las opciones de control para ejecutar la regla permite configurarla para que una vez que encuentre el patrón que busca, ya no continúe la búsqueda. Esto facilita descripción de la regla, entonces sólo es necesario indicar que acepte una serie de palabras y espacios seguidos. La limitante es en realidad indicar dónde finaliza el título, se sabe que un título se escribe en un sólo renglón, sin embargo, si es muy largo, continuaría en otros renglones, saber cuántos renglones componen al título es muy complicado, por lo tanto se deben aceptar como máximo dos renglones.

Autores

La forma de presentar un autor es el tipo de información que más varía entre las distintas publicaciones que se toman en cuenta. El nombre de un autor se caracteriza por ser una secuencia de al menos dos palabras que inician con letra mayúscula. Sin embargo, hay muchas consideraciones y variantes que se deben agregar para poder identificar correctamente un autor o secuencia de autores. Hay dos maneras en la que los autores de un artículo pueden presentarse, un autor en cada renglón o una serie de autores separados por espacios o comas. En la primera presentación la regla no se complica, ya que es relativamente fácil encontrar una pequeña secuencia de palabras que inician en mayúscula, caso distinto a etiquetar una secuencia de autores, ya que cuando se presenta este caso es común que en la secuencia no se encuentren únicamente los autores, también es común que enseguida de cada autor se encuentre la dirección de correo electrónico del autor o bien la universidad u organización a la que pertenece, además de que al considerar una secuencia larga de palabras es posible confundir el nombre de un autor con el nombre de una organización o el nombre de un proyecto, el nombre de una ciudad, etc.

A pesar de las variaciones de estilo, un autor siempre se encuentra al inicio de un renglón, por eso es importante identificar primero un salto de línea, ya que se sabe que éste existirá porque anterior al autor se encuentra al menos el título del artículo. Después puede haber varios espacios, pero enseguida se encontrará el nombre completo de un autor. El autor al menos debe contener

el nombre y apellido del autor, sin embargo, también puede estar un segundo nombre del autor o quizás su letra inicial con un punto. Otra variación es que el nombre puede tener el prefijo “de”, que no inicia en mayúscula y que en otros idiomas puede aparecer con “van” o “di” entre otros.

El nombre o apellidos por sí sólo pueden no ser una única palabra, este es el caso de los nombres que tienen acento, GATE detecta el acento como un carácter separado de la letra que lo lleva, de modo que el nombre del autor queda separado en dos palabras divididas por su acento. Este mismo caso se da cuando dos apellidos se unen por un guión. El nombre completo de un autor puede continuarse con un símbolo o número, que sirve para identificar una nota sobre los datos del autor, enseguida se debe encontrar un coma o un salto de línea. En el caso de encontrar una coma se consideran las posibilidades de encontrar más autores o datos relacionados con ellos. Para el primer nombre de un autor se considera que no todas deben estar escritas en minúscula, debe tratarse de una palabra diferente a cualquier verbo o alguna de sus conjugaciones.

Resumen

El resumen es la parte de un artículo, que se encuentra normalmente después del título y sus autores. Generalmente inicia con la palabra “Abstract” y continua con algún signo de puntuación, en otros casos, la misma palabra se utiliza como el título de esta parte del documento. También hay ocasiones donde el resumen no está titulado y no inicia con ninguna palabra en especial.

El conjunto de palabras y renglones que forman el resumen puede contener una serie de palabras de cualquier tipo, números, símbolos o signos de puntuación por lo tanto marcar el resumen es un problema que trata de encontrar dónde inicia y dónde finaliza esta parte del documento. Ejecutar una regla que trate de encontrar cualquier serie de caracteres y que sólo se diferencie por el inicio y final de la serie, sería muy costoso. La mejor opción para encontrar el resumen es marcar únicamente la palabra “Abstract” seguida por los símbolos o signos de puntuación característicos del inicio de esta sección. En una etapa posterior se podrá obtener la sección más cercana a ésta, como una anotación “Sección” o “Palabras clave”, que marcarán el final del resumen.

Palabras clave

Las palabras clave son una serie de palabras que se presentan en el documento como las palabras más relevantes del artículo. Estas palabras son una parte del documento que se indica con las palabras “Keywords” o “Index Terms” como un pequeño título o como el inicio del renglón donde se enlistan las palabras las cuales pueden, estar separadas por cualquier signo de puntuación, escritas en minúscula o con la primera letra en mayúscula, formar más de un renglón y terminar con un salto de línea.

La cantidad de renglones de palabras clave es muy variable y ya que no hay una marca especial

que indique el fin de éstas, la única forma de identificar dónde terminan es identificando dónde inicia la siguiente sección, la cual estará marcada por la anotación "Sección", sin embargo, en algunos casos la siguiente sección no se encuentra marcada por ningún título y para estos casos se debe tomar la precaución de limitar la cantidad de renglones que componen esta anotación, según las pruebas realizadas está cantidad debe ser no más de cuatro.

3.1.2. GATE embebido

El API de GATE para Java brinda la posibilidad de que el usuario utilice los módulos de GATE que mejor se adecúen a las necesidades de cada aplicación, sin la necesidad de utilizar todas las herramientas con las que cuenta GATE. Para obtener los datos del artículo no es necesario utilizar todos los módulos que proporciona GATE, básicamente se utilizan la mayoría de los módulos que contiene una aplicación de tipo ANNIE.

El procedimiento general para que GATE procese los documentos mediante Java, es el siguiente:

- Indicar las direcciones donde se encuentran los recursos de GATE.
- Iniciar GATE.
- Crear un nuevo Corpus que formará el conjunto de documentos que analizará GATE.
- Agregar los archivos al Corpus.
- Indicar los módulos de GATE que procesaran los documentos, estos son:
  - *Document Reset PR*: Recupera el documento a su estado original, quitando cualquier tipo de anotación previa. En este módulo también se eliminan las partes del documento que no son texto como imágenes u otros elementos que dividan el documento, en algunas ocasiones estos elementos pueden convertirse en símbolos desconocidos. También elimina el formato de varias columnas del documento, dejándolo en una sola columna, lo que desafortunadamente puede generar problemas en documentos donde la distribución de columnas es muy variada.
  - *DefaultTokeniser*: Segmenta el documento en palabras, números, símbolos y espacios, etc. Este módulo es muy importante porque genera gran parte de las anotaciones que necesitan las reglas Jape que se utilizarán más adelante para extraer la información que se busca en estos documentos.
  - *DefaultGazeteer*: Identifica entidades nombradas basado en listas de documentos.
  - *ANNIETransducer*: Identifica entidades y eventos, ejecutando las reglas que el sistema contenga, es en este módulo que las reglas para identificar título, autores, resumen y palabras clave se ejecutan.

- Ejecutar los módulos sobre el Corpus.
- Extraer etiquetas de los documentos.

Los pasos anteriores se pueden seguir para ejecutar las reglas que se implementen en Jape para obtener la información requerida de cada artículo. Los módulos que procesan el documento antes del *ANNIETransducer* sirven para preparar al documento para que al ejecutar existan las anotaciones necesarias para que las reglas identifiquen correctamente los datos que buscan.

3.1.3. Extracción de anotaciones

Una vez que los documentos se procesan con GATE y se generan las anotaciones correspondientes de cada documento, el paso siguiente es extraer las anotaciones que realmente son útiles del documento. El proceso de extraer información se compone de varias etapas y durante ellas se generan varias anotaciones, en una de ellas se generan anotaciones muy básicas, como la segmentación de palabras, números, símbolos, en otra etapa se utilizan estas anotaciones para obtener nuevas anotaciones más complejas de identificar.

Para obtener las anotaciones que si tienen interés, se recorren cada una de las anotaciones, las cuales tienen un tipo, un inicio y un final. El tipo se refiere al nombre de la anotación, por ejemplo “Autores” o “Sección”, mientras que el inicio y el final se refieren al número de carácter con respecto a los caracteres de todo el documento dónde inicia y finaliza la anotación.

En el caso de no encontrar alguna de las anotaciones se toman algunas medidas. Para el caso de los autores no es necesario hacer nada, ya que de no encontrarse no significaría que el documento quede inutilizado para relacionarse con otro documento. Claramente no habría información sobre al autor y no se podría hacer la relación con otro autor, sin embargo, la relación también puede hacer por medio de su resumen o palabras clave. En caso de no encontrar un título en el documento, tampoco sería un problema grave ya que no se necesita para obtener otros datos y tampoco impide hacer la relación del artículo con otros autores u otros artículos.

Para encontrar el resumen es necesario encontrar la siguiente sección o las palabras clave, que generalmente se encuentran después del resumen. Por lo tanto la razón para identificar la siguiente sección es que puede servir para encontrar el final del resumen en caso de que las palabras clave no se encuentren. El conjunto de palabras clave es una anotación difícil de repetir, normalmente se encuentra sólo una vez, pero basta con tomar la primera que se encuentre para eliminar otras anotaciones que en realidad no son la información buscada.

El contenido de la anotación “Sección”, no es información que interese al sistema, únicamente se utiliza como auxiliar para obtener otros datos, el problema de no encontrarla se presenta cuando tampoco se encuentra la anotación de palabras clave, o se encuentre arriba del resumen, en este

caso no hay más opción que tomar una cantidad fija de caracteres después de la anotación que marca el inicio del resumen.

Otro problema puede presentarse cuando no se encuentra la anotación del resumen. La razón para no encontrarlo puede deberse a que esta sección del documento no se encuentra o también porque esta sección no está marcada por un título o un inicio bien definido. En tal caso se puede recurrir a identificar el carácter final del autor que aparezca más adelante en el documento, el cual es muy probable que se encuentre muy cerca de donde debería encontrarse la anotación del resumen.

Cabe recordar que cualquier anotación es una serie de palabras, símbolos, números o espacios que se ajustan al patrón que busca una regla. En el caso del título y el resumen, la anotación identifica la información sin algún tipo de dato extra, pero en el caso de los autores la anotación identifica un conjunto de datos con el nombre del autor, comas, universidades y correos electrónicos, mientras que en las palabras clave, la anotación identifica el conjunto de palabras, con comas y espacios.

Para identificar cada autor y cada palabra principal de un artículo se necesita pasar la anotación recuperada por un filtro que elimine los datos que no interesan. Este filtro se debe desarrollar en una clase de Java, con el fin de disminuir el costo de procesamiento de cada artículo en GATE, ya que de adoptar un enfoque para que las anotaciones entreguen los autores y palabras por separado, se tendrían que agregar varias reglas que generarían muchas anotaciones adicionales de cadenas de palabras similares al nombre de un autor, que incrementarían potencialmente el tiempo de procesamiento de cada documento.

La figura 3.2 muestra el procesamiento que se hace a las anotaciones que se generan con GATE para obtener los datos finales de las anotaciones. En el resumen o en otras partes del documento que se quieran procesar, se deben obtener las sentencias que podrían contener información importante, tales como las que mencionan algo acerca de las palabras clave del documento.

3.2. Almacenamiento de información

En cada etapa del procesamiento de datos se obtiene información que es utilizada en etapas posteriores para generar más información. Los primeros datos que deben almacenarse son los artículos científicos que se analizan, y del procesamiento de los documentos surge la información más importante de cada artículo que sirve en las siguientes etapas de sistema. Esta información no puede ser eliminada porque obtenerla requiere tiempo y cada vez que se requiere hacer nuevas relaciones, estos datos serán utilizados de nuevo. La información que se ha mencionado es conveniente almacenarla en una base de datos. Existen otros tipos de datos que surgen del procesamiento de los datos almacenados, los cuales es más conveniente tenerlos en una ontología.

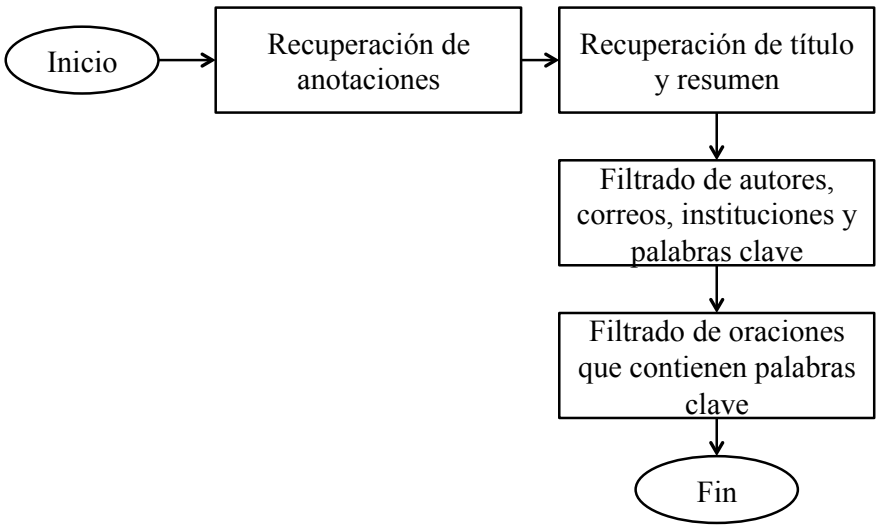

Figura 3.2: Procesamiento de anotaciones para la extracción de datos de documentos

En esta sección se presenta el diseño de la base de datos que utiliza el sistema, el tipo de datos que almacena y la utilidad que tienen éstos. La base de datos es una parte my importante en el sistema ya que almacena la información que se obtiene de cada documento la cual es utilizada después, y por ello es necesario obtenerla rápidamente y tenerla bien organizada.

Por otra parte se presenta el diseño de la ontología que se utiliza para generar información a partir del procesamiento de artículos científicos. A diferencia de la base de datos, la ontología no sólo almacena información, sino que ayuda a generar información, por ello se utiliza un diseño diferente al de la base de datos, adecuado a la nueva información que se genera de cada artículo y autor.

3.2.1. Diseño de la base de datos

La base de datos diseñada para el sistema pretende almacenar los datos extraídos de los artículos. En la figura 3.3 se puede observar la base de datos que se compone de tres entidades *Authors*, *Documents* y *Sentences*. La entidad *Author* almacena la información de cada autor de un artículo, se compone de cuatro atributos, el primero llamado *IdAuthor* sirve para identificar cada autor en la base de datos y relacionarlo con otras entidades, los datos que se guardan sobre un autor son nombre, correo electrónico e institución a la que pertenece y son almacenados en los atributos *Name*, *Email* e *Institute* respectivamente.

La segunda entidad *Documents* almacena los datos de propio artículo tales como título, resumen y palabras clave, los atributos que contienen estos datos son *Title*, *Abstract* y *Keywords* respectivamente. La relación que existe entre la entidad *Authors* y *Documents* es N a M porque

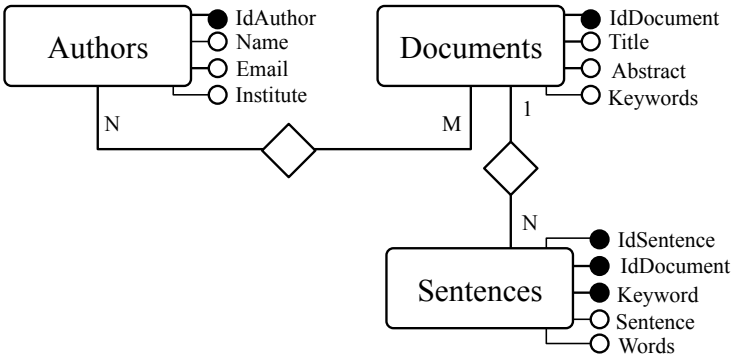

Figura 3.3: Diagrama entidad-relación de la base de datos

un mismo autor puede escribir muchos artículos y un artículo puede ser escrito por varios autores. El atributo *IdDocument* sirve para identificar cada documento y formar la debida relación con la entidad *Authors*.

Por otra parte, *Documents* también está relacionada con la entidad *Sentences* que representa una sentencia que pertenece exclusivamente a un documento. Una sentencia del documento es almacenada cuando ésta contiene una de las palabras clave *keywords* del propio documento. Otra información importante de la sentencia es el conjunto de palabras relacionadas con la palabra clave asociada a la sentencia. El conjunto de palabras debe estar formado por las palabras más significativas de la oraciones, tales como sustantivos, verbos o adjetivos. Para conocer el conjunto de palabras, la sentencia debe ser procesada, de forma que sea posible identificar las categorías gramaticales de cada palabra y su relación con la palabra clave. Este proceso puede realizarse obteniendo el árbol sintáctico de la oración. La entidad *Sentences* tiene cinco atributos, *IdSentence* sirve para identificar el número de oración en el documento, *IdDocument* ayuda a asociar la sentencia al documento que pertenece, *Keyword* indica qué palabra clave se encuentra en la sentencia, *Sentence* almacena la propia sentencia y *Words* almacena las palabras más significativas asociadas a la palabra clave en la sentencia.

3.2.2. Diseño de la ontología

La ontología tiene como objetivo representar las relaciones semánticas entre artículos científicos y sus autores. En la figura 3.4 se observan las seis clases que componen la ontología, así como las relaciones entre ellas, mientras que la figura 3.5 muestra los atributos de cada clase. Las clases que integran la ontología se explican a continuación:

- *Document*: la clase *Document* modela la información sobre un artículo científico cuyos atributos son: identificador, titulo, palabras clave y resumen; los cuales son representados en los

atributos *IdDocument*, *Title*, *KeyWords* y *Abstract* respectivamente. Un artículo tiene uno o más autores, los cuales son modelados mediante la clase *Author*.

- *Author*: la información de un autor es el nombre, el correo electrónico y la institución a la que pertenece. Estos datos son modelados en los atributos de la clase, llamados *Name*, *Email* e *Institute* respectivamente. El identificador de cada autor es representado en el atributo *IdAuthor*.

- *WordsGroup*: esta clase modela la información sobre un grupo de palabras contenida en un artículo científico. El tipo de palabras que forman el grupo son adjetivos y sustantivos, los cuales son representados en los atributos *Adjectives* y *Nouns* repectivamente, además cuenta con *IdWordsGroup* para identificar cada grupo de palabras.

- *GeneralGroup*: esta clase es una subclase de *WordsGroup* y representa un grupo general de palabras asociado a una palabra clave del documento. Cada palabra clave genera un grupo general, la cual es representada en el atributo *KeyWord*. Un grupo general, le pertenece al documento que se analizó para obtener las palabras asociadas a su palabra clave, sin embargo, este mismo grupo o al menos parte de él, puede contenerse en otro documento. Esta clase cuenta con un atributo llamado *Concurred* y sólo puede tomar los valores "verdadero" o "falso". Cuando un grupo general sólo le pertenece a un documento, el atributo *Concurred* toma el valor "falso", en cambio si el grupo general es un grupo de palabras que aparece a más de un documento, el atributo *Concurred* toma el valor "verdadero".

- *SpecificGroup*: Tres documentos o más, que coinciden en poseer un mismo grupo general de palabras, también pueden contener otro grupo de palabras que tiene relación con la misma palabra clave del grupo general de palabras. A este segundo grupo se le llama grupo especifico y es representado por esta clase. La clase *SpecificGroup* hereda de la clase *WordsGroup*, por lo tanto también cuenta con adjetivos y sustantivos, pero en este caso representan las palabras del segundo grupo que coincide en los documentos. Ya que un grupo especifico está relacionado con una palabra clave y existe el grupo general de palabras que en primer lugar relaciona los documentos que contienen un grupo especifico, se dice que *SpecificGroup* tiene un *GeneralGroup*.

- *Classification*: Un documento puede contener varias palabras clave y por lo tanto varios grupos generales asociados a él. Además un mismo grupo de palabras, ya sea general o especifico puede estar en uno o más documentos. Las relación entre grupos de palabras y documentos es representada en una clasificación de documentos, y está modelada en esta clase. De modo que, *Classification* tiene un *Document* y un *WordsGroup*, el cual puede ser un *GeneralGroup* o un *SpecificGroup*. Un documento puede estar asociado a varias clasificaciones, si se quisiera tomar un segmento de documentos asociados a un grupo de palabras, sería necesario saber qué relación entre documentos y grupo de palabras es más representativa que otras, está distinción es representada por el atributo *Distinctive* y el grado de representatividad lo contiene el atributo *DistinctiveDegree*. Para identificar la clase está el atributo *IdClassification*.

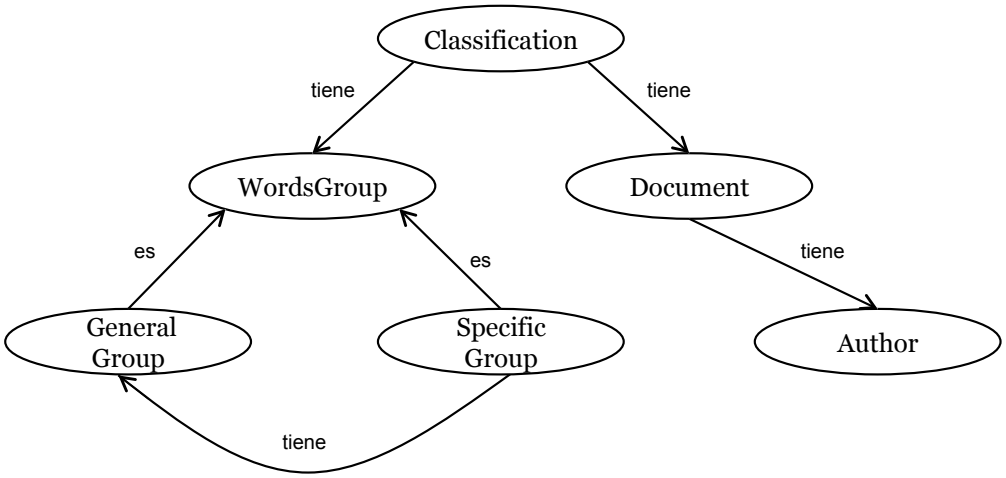

Figura 3.4: Diagrama de la ontología

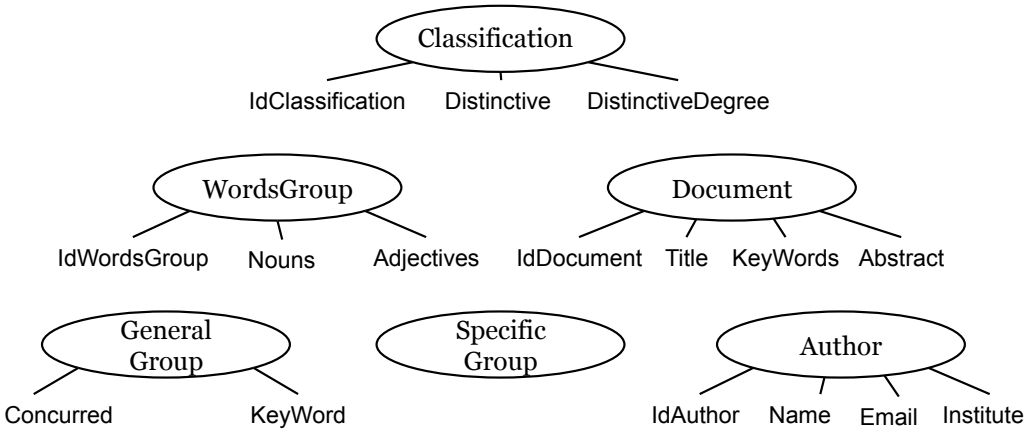

Figura 3.5: Clases de la ontología con sus atributos

En la figura 3.6 se observa un diagrama de Venn con tres conjuntos, donde cada uno representa el conjunto de palabras asociadas a una misma palabra clave pero en documentos distintos. La intersección de los tres conjuntos forma lo que se ha llamado un grupo general, mientras que la intersección entre dos conjuntos eliminando el conjunto de grupo general forma un grupo específico.

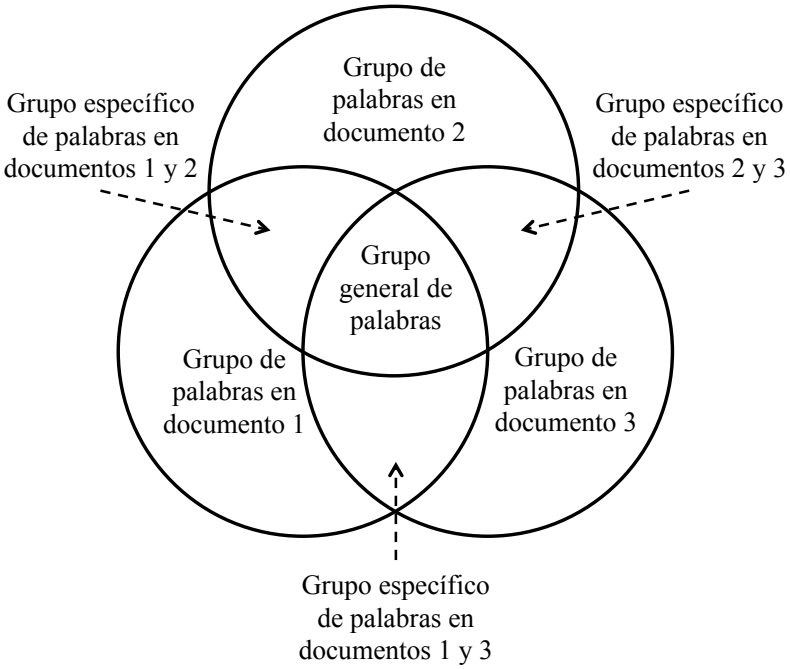

Figura 3.6: Grupo general y grupos específicos formados por palabras de diferentes documentos

3.3. Relaciones semánticas

La siguiente etapa en el procesamiento de la información es obtener relaciones semánticas entre la información que se generó de la extracción de datos de los artículos científicos. Esta sección presenta las técnicas que se utilizan para obtener relaciones semánticas entre artículos científicos y sus autores. Las relaciones se guardan en la ontología y en esta sección se presentan los métodos para hacer el poblado automático de la ontología.

Uno de los métodos utilizados para relacionar un artículo hace la comparación de algunas sentencias en un par de artículos. La caracterización de las sentencias se forma a partir de los árboles sintácticos que se obtienen de las frases del texto, en las siguientes secciones se detalla el método empleado para obtener tales árboles, los parámetros que se toman en cuenta para analizar o no una frase, así como las métricas que se siguen para estimar un grado de similitud entre sentencias.

Con los resultados de la comparación de sentencias, se obtiene información que sirve para relacionar un conjunto de artículos y autores. A continuación se presentan los métodos para formar nuevas relaciones con otros artículos, autores con otros autores y artículos con autores.

Los datos que se utilizan para generar relaciones semánticas son las palabras clave del artículo, sentencias de resumen, entre otras. Otras relaciones pueden ser descubiertas mediante consultas a

la ontología, las cuales generan información sobre la relación que tienen varios artículos y varios autores, quizás sobre un tema determinado.

3.3.1. Comparación de oraciones

Los métodos para encontrar relaciones semánticas entre autores de artículos y sus tópicos de investigación son abordados desde diferentes perspectivas, uno de ellos es la similitud de textos en artículos científicos. El método propuesto toma en cuenta principalmente los principios de Micol [27] y Wang [29]. Nuestro trabajo propone hacer la comparación entre arboles sintácticos construidos mediante el *Parser Stanford* [18], el cual es un analizador sintáctico probabilístico capaz de resolver la estructura gramátical de una sentencia.

A diferencia de otras propuestas, este método además de encontrar coincidencias entre los nodos de árboles y evaluar la similitud entre ellos, durante la comparación de sentencias, el parser extrae un árbol de cada sentencia y los cuales son explorados en varias ocasiones, iniciando la exploración de ramas desde diferentes hojas.

El número de recorridos está determinado por el número de coincidencias de palabras entre las oraciones. La comparación de los árboles sintácticos se lleva a cabo recorriendo uno a uno los nodos de ambos árboles tratando de encontrar coincidencia entre las categorías gramaticales que representa cada nodo.

La propuesta está enfocada a encontrar la similitud de sentencias mediante la comparación de sus correspondientes árboles sintácticos. La estimación de la similitud depende del tipo y número de nodos similares entre los nodos. Todos los nodos son importantes, incluso aquellos que no son similares. Incluso si todos los nodos en un árbol A están en un árbol B, no es razón suficiente para obtener el máximo grado de similitud. Dos sentencias tienen el máximo grado de similitud cuando todos los nodos en A están en B.

Descripción del método

En la comparación de dos sentencias, por requisito una de ellas debe contener al menos dos palabras que puedan ser emparejadas con palabras en la otra sentencia. Dos palabras son emparejadas cuando las palabras sin su variedad morfológica son las mismas. El método puede ser usado para encontrar la similitud entre un texto y una sentencia o varias sentencias, pero es necesario evaluar qué sentencias del texto cumplen con el requisito para que el método pueda ser usado.

En la figura 3.7, el diagrama a bloques describe la operación del método propuesto. Una vez que se han seleccionado un par de sentencias, el siguiente paso es obtener el árbol sintáctico de cada sentencia, este procedimiento permite conocer si las sentencias satisfacen el requisito para

ser comparadas. Hay que tomar en cuenta que es posible que existan palabras con una variedad morfológica diferente, pero con el mismo significado. Una vez que los árboles han sido obtenidos, se pueden descubrir los términos más significativos de cada sentencia como sustantivos, verbos o adjetivos, los cuales pueden ser usados para descubrir el número de pares de palabras emparejadas en la sentencia. Es importante mencionar que una palabra diferente a un sustantivo, verbo o adjetivo no debe ser considerada para hacer el emparejamiento de palabras en las sentencias.

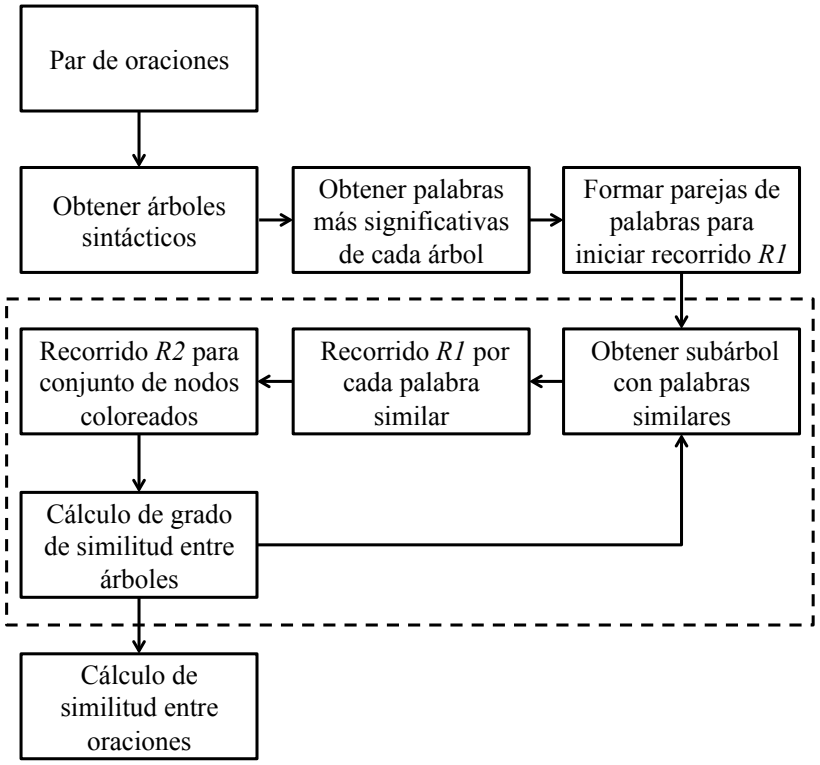

Figura 3.7: Diagrama a bloques del método para comparar oraciones

En un par de sentencias puede haber más de dos palabras en común. El siguiente paso es construir pares de palabras, haciendo la combinación de palabras en común. El número de comparaciones entre dos árboles es el mismo que el número de pares construido por la combinación de palabras. Los bloques en la figura 3.7 dentro del recuadro punteado, son los bloques que se ejecutan por cada par de palabras comunes en las dos sentencias. La comparación de árboles se ejecuta en cuatro etapas, la primera consiste en obtener los subarboles más pequeños que contienen el par de palabras emparejadas seleccionadas para hacer la comparación, en la segunda y tercera etapa se comparan los árboles mediante los recorridos llamados *R1* and *R2* respectivamente y en la cuata etapa se calcula el grado de similitud entre árboles. Finalmente, se puede calcular la similitud entre sentencias.

Comparación entre árboles sintácticos

A partir de cada palabra significativa en ambas sentencias, se genera un lista formada mediante la combinación de pares de palabras es decir, (p_1, p_2), (p_1, p_n), ..., (p_2, p_3),..., (p_{n-1}, p_n). De cada árbol es necesario encontrar los subárboles más pequeños que contiene lo pares de palabras formados. Estos subárboles usualmente no son los mismos que los árboles completos de la sentencia, de modo que esto permite eliminar las partes menos significativas de las oraciones. Los subárboles son llamados A y B y se selecciona A como el que tiene mayor cantidad de nodos.

La comparación de oraciones se compone de dos procesos, en el primero, los árboles son coloreados de acuerdo a la similitud entre los nodos de cada árbol. En el segundo, los nodos son pesados de acuerdo a la categoría gramatical asignada al nodo. Finalmente la ecuación 3.1 obtiene una proporción de similitud entre todos los nodos similares en los árboles.

Recorrido de los árboles

El recorrido de los dos árboles que se comparan, se realiza simultáneamente. Para iniciar, se escoge una de las palabras que aparecen en ambas oraciones, y se ubica el nodo donde aparecen, a partir de estos nodos inicia el recorrido en su correspondiente árbol. El primer objetivo en esta parte del método es colorear los nodos similares que se encuentren en los caminos del árbol A y B que van desde la palabra escogida hasta la raíz de cada árbol, a este tipo de recorrido se le nombra *R1*.

Dos nodos similares son los que tienen una categoría gramatical similar, es decir que, un nodo que representa un verbo en forma simple es similar a un nodo que representa un verbo en gerundio. Las categorías gramaticales indicadas por su abreviación [41], que se toman como similares se indican en el cuadro 3.1.

Cuando dos nodos similares son encontrados, estos nodos son coloreados y la comparación puede continuar. El siguiente par de nodos a comparar son los primeros nodos ancestro de los nodos coloreados. La figura reffigura1a muestra este caso (arco1), señalando el nodo ancestro en A que puede ser comparado con el nodo en B.

Si los nodos ancestro no son similares, los nodos no se coloran y la siguiente comparación será entre el mismo primer nodo ancestro del nodo coloreado en B y el segundo nodo ancestro del nodo coloreado en A (figura 3.8a, arco 2). Si estos, en A y B son similares, ellos se colorean y la siguiente comparación es entre los primeros ancestros de los nuevos nodos coloreados, pero si los nodos no son similares, la siguiente comparación es entre el tercer nodo ancestro en A y el mismo nodo ancestro en B (figure 3.8a, arco 3).

Las comparaciones pueden continuar sin colorear hasta el tercer nodo ancestro en A, o hasta

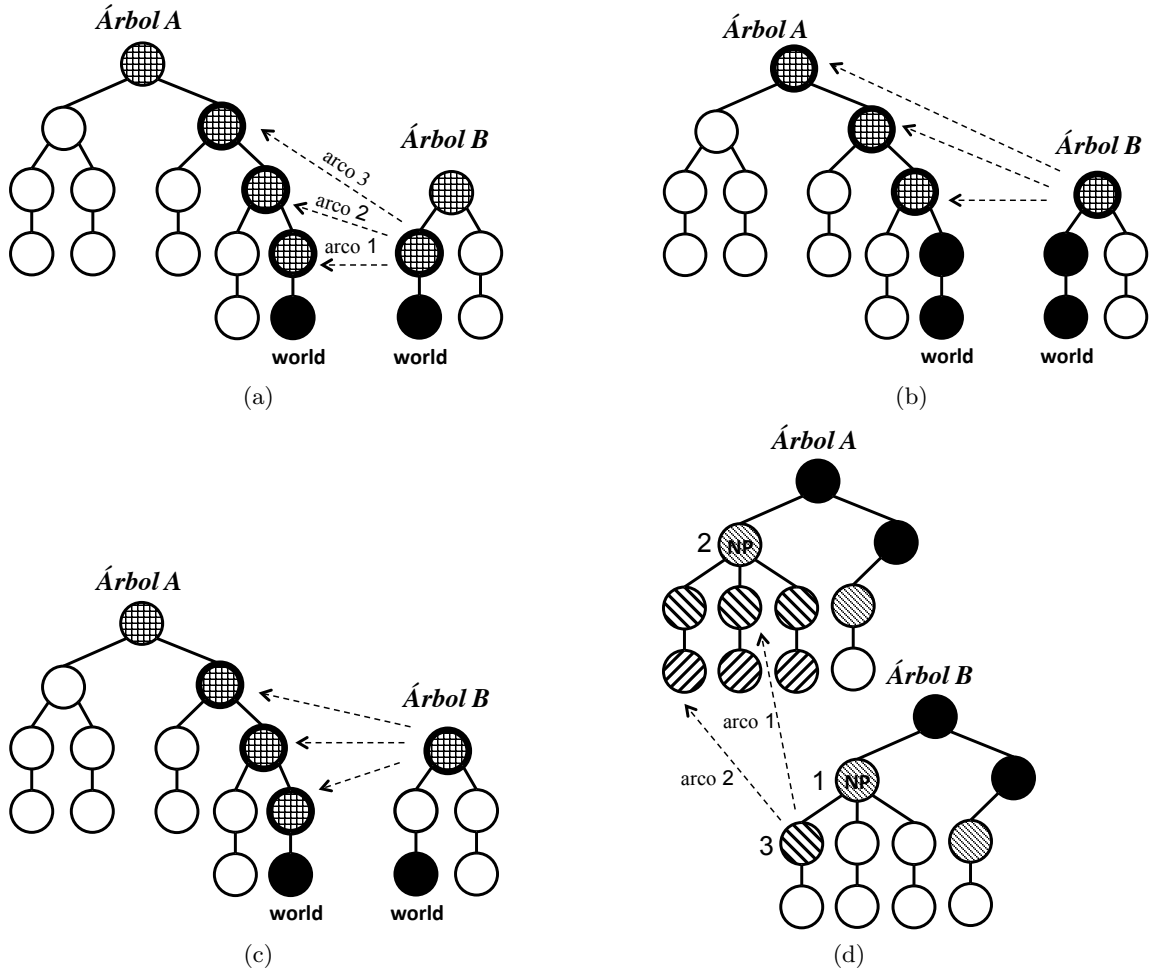

Figura 3.8: Exploración de árboles.

llegar a la raíz, sin cambiar el nodo comparado en B. Esto es ilustrado en la figura 3.8a, donde el nodo en B es comparado con máximo tres nodos ancestro en A para encontrar un nodo similar.

Cuando los nodos ancestro son similares, ambos nodos son coloreados y el método continua con la siguiente comparación como es mostrado en la figura 3.8b. El método establece que cuando la comparación alcanza al tercer nodo ancestro en A sin encontrar nodos similares, el nodo en B, no es similar con ningún nodo en A , como se muestra en la figura 3.8c. El número de ancestro en B puede incrementar de nuevo, si hay el mismo número de condiciones descritas, hasta alcanzar la raíz del árbol.

Este proceso es repetido, pero esta vez usando la segunda palabra aparecida en ambas sentencias. Cuando el recorrido de ambos caminos ha finalizado, tenemos un primer conjunto de nodos

coloreados en el árbol A, y se sabe que ellos tienen un correspondiente nodo similar en el árbol B. A partir de cada nodo en el conjunto, inicia una nueva exploración, llamada *R2*. Esta exploración intenta encontrar nuevos nodos similares entre los nodos hijos de nodos coloreados en A y B, como se muestra en la figura 3.8d donde se observan los nodos 1 y 2 los cuales son similares entre sí y el arco 1 que señala al nodo 3 que es hijo del nodo 1, comparandose con los hijos del nodo 2. Si el nodo 3 y alguno de los hijos del nodo 2 son similares, estos nodos se colorean, y la comparación continúa con los demás hermanos del nodo 3 y los nodos hijos no coloreados del nodo 2. Sin embargo, si el nodo hijo de un nodo coloreado en B no encuentran un nodo similar en los nodos hijos del nodo coloreado en A, el nodo hijo del nodo coloreado en B puede buscar nodos similares en los nodos nietos del nodo coloreado en A, siempre y cuando el padre del nodo que se compara no haya sido coloreado antes, tal como se muestra en la figura 3.8d, donde el arco 2 señala que el nodo 3 también puede ser comparado con los nodos nietos del nodo 2.

Cuando se encuentran nodos similares en la comparación entre cada hijo del nodo en B y los nodos hijos del nodo en A, estos nodos además de ser coloreados, también son seleccionados para formar un nuevo conjunto de nodos. Enseguida se da inicio al siguiente recorrido, en busca de hijos similares de los nuevos nodos coloreados, que finalmente podría llegar hasta las hojas del árbol.

En la figura 3.9 se muestra un ejemplo de árboles A y B, cuyos nodos fueron coloreados aplicando los recorridos *R1* y *R2* a partir de las palabras *tiger* y *world*. En el método los nodos simplemente son coloreados, sin embargo, en la figura los nodos son coloreados en un color diferente, para diferenciar el tipo de recorrido en el que cada nodo fue coloreado. Los nodos coloreados con un color diferente a negro son resultado de usar *R2* en varias ocasiones.

Etiqueta	**Etiqueta**
ADJP	ADP
NP	PP
S	SBAR, SBARQ
SINV	SQ
VP	WHADVP
WHNP	WHPP
CC	POS
CD	PRP
DT	RB, RBR, RBS
EX	RP
FW	SYM
IN	TO
JJ, JJR, JJS	UH
LS	VB, VBD, VBG, VBN, VBP, VBZ
MD	WDT
NN, NNS, NNP, NNPS	WP
PDT	WRB

Cuadro 3.1: Grupos de etiquetas con similitud gramatical.

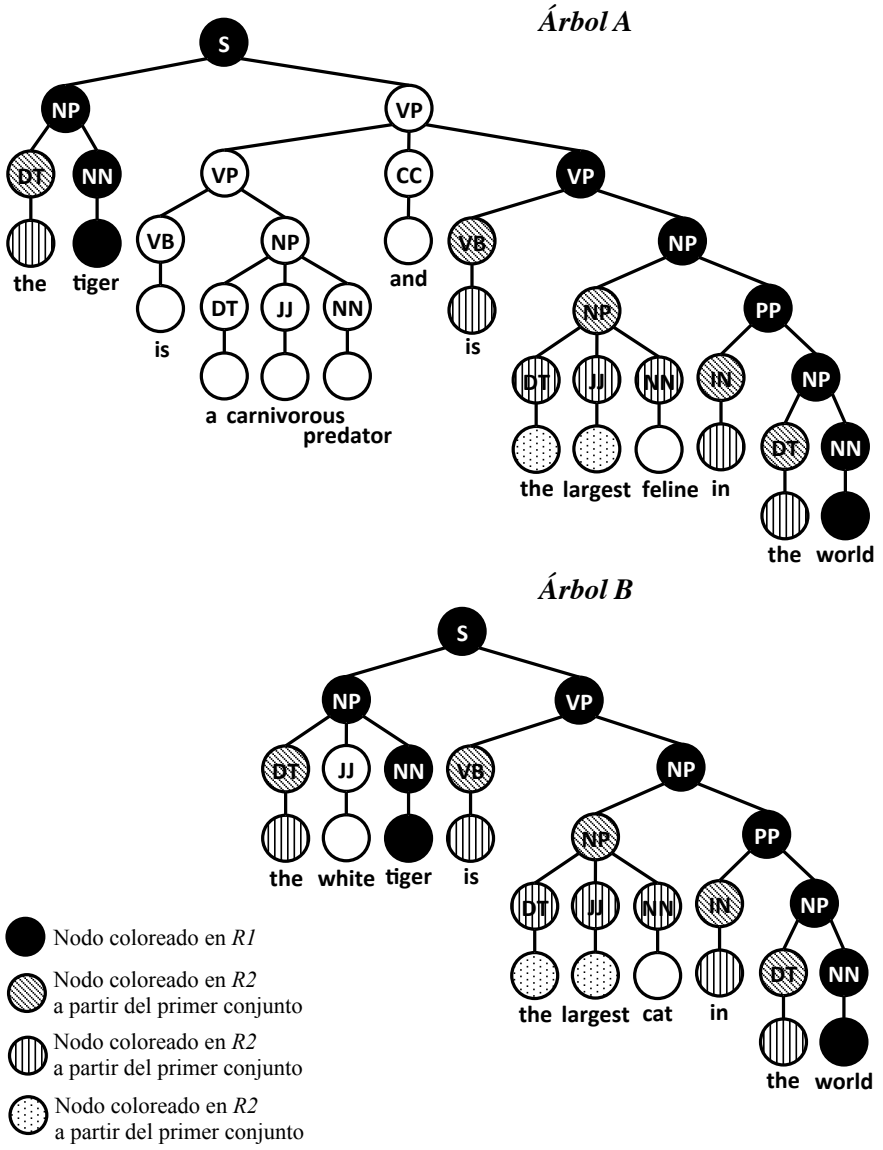

Figura 3.9: Recorrido y coloreado de árboles

Similitud entre árboles

Cuando el recorrido de los árboles ha finalizado, si las sentencias son completamente iguales, todos los nodos deberían ser coloreados y el grado de similitud es uno. El valor mínimo de similitud es cero, pero este valor no puede ser dado a ningún par de árboles, ya que éstos deben tener al menos las dos palabras para iniciar el proceso de comparación. Es importante decir que encontrar nodos similares que representan un verbo, sustantivo o adjetivo no es tan importante como encontrar

nodos similares entre nodos que representan otro tipo de categoría gramatical.

El valor de cada nodo (coloreado o no) es tomado en cuenta para obtener la similitud final entre los árboles A y B. Los valores que se dan a los diferentes grupos de nodos son diferentes entre sí, debido a que en la comparación propuesta no se intenta encontrar frases iguales, si no similares, es decir que sus sustantivos y verbos sean utilizados en una estructura sintáctica similar, independientemente del tiempo, género o número. El cuadro 3.2 muestra el valor que se le da a cada categoría gramatical [41] señalada por la etiqueta que se le asigna al nodo.

Etiqueta	**Peso**
Palabra	4
ADJP, ADVP, NP, PP, VP	3
WHADVP, WHNP, WHPP	3
NN, NNS, NNP, NNPS	2
VB, VBD, VBG, VBN, VBP, VBZ	2
Otras	1

Cuadro 3.2: Peso del nodo, de acuerdo a su categoría gramatical.

Para obtener el grado de similitud S_t entre el árbol A y B, se recorren ambos árboles y se aplica la ecuación 3.1, donde W_{nC} representa el peso de cualquier nodo coloreado en el árbol A (o B, debido a que son los mismos), W_{nA} representa el peso de cualquier nodo coloreado o no del árbol A y W_{nB} representa el peso de cualquier nodo coloreado o no del árbol B.

$$S_t = \frac{\sum_{k=1}^{r} W_{nCk}(\sum_{i=1}^{p} W_{nAi} + \sum_{j=1}^{q} W_{nBj})}{2\sum_{i=1}^{p} W_{nAi} \sum_{j=1}^{q} W_{nBj}} \tag{3.1}$$

La similitud de la oración, está determinada por la ecuación 3.2, que es el promedio de grados de similitud que se obtiene entre cada par de árboles, los cuales se forman de la combinación de dos sustantivos que aparecen en ambas oraciones, que sirven para iniciar los recorridos *R1* de los árboles, con cada una de estas palabras.

$$S_p = \frac{\sum_{i=1}^{C} S_{ti}}{C} \tag{3.2}$$

3.3.2. Poblado automático de la ontología

Los datos que son extraídos en etapas anteriores son útiles para poblar las clases *Document* y *Author* directamente en la ontología diseñada, sin embargo, obtener la información para poblar las

clases *GeneralGroup*, *SpecificGroup* y *Classification* requiere procesar algunos datos y hacer algunas consultas a la base de datos y a la ontología misma. El procesamiento de datos de un documento puede desembocar en la creación de grupos generales o específicos nuevos, y siempre debe generar clasificaciones del documento ya sea con los nuevos grupos o con grupos formados anteriormente. A continuación se listan los conceptos que se utilizan durante el poblado de la ontología:

- **GG**: Grupo General, se refiere a un grupo de palabras que puede formar un individuo de la clase *GeneralGroup*.
- **GGnC**: Grupo General no Concurrente, se refiere a un grupo general de palabras que se forma a partir de un documento nuevo, y que es utilizado para formar nuevas relaciones con los otros grupos de palabras de la ontología.
- **GGp**: Grupo General probable, se refiere a un grupo general de palabras que tiene una probable relación con otro grupo general, debido a que comparten la misma palabra clave, pero aun no se conoce si las palabras del grupo son utilizadas en oraciones similares.
- **GGCN**: Grupo General Concurrente Nuevo, se refiere a un grupo general nuevo que se forma del descubrimiento de una relación de dos grupos de palabras, lo que lo convierte en concurrente.
- **GE**: Grupo Específico, se refiere a un grupo de palabras que puede formar el individuo de la clase *SpecificGroup*.
- **GEp**: Grupo Específico probable, se refiere a un grupo específico de palabras que tiene una probable relación con otro grupo específico, debido a que comparten un mismo grupo general, pero aun no se conoce si las palabras del grupo son utilizadas en oraciones similares.
- **GEN**: Grupo Específico Nuevo, se refiere a un grupo específico nuevo de palabras que se ha confirmado que las palabras que lo componen tienen relación en las oraciones donde aparecen.
- **CLS**: Clasificación, se refiere a un individuo de la clase *Classification*.

La figura 3.10 muestra el diagrama de flujo general poblar grupos generales y específicos así como clasificaciones. A partir de una palabra clave se obtienen las palabras más significativas relacionadas con ella en las oraciones del documento. Las palabras relacionadas junto con la palabra clave forman un grupo de palabras general no concurrente (GGnC) propio del documento. Por cada palabra clave se forma un GGnC que junto con su documento forman una clasificación (CLS). El GGnC permite guardar todas las palabras asociadas a su palabra clave aun cuando éstas no coincidan con otros grupos en la ontología, permitiendo así que otros documentos que puedan ser analizados en el futuro encuentren coincidencias con este documento. Un grupo general probable (GGp) es creado por la intersección de dos GGnC y es un potencial candidato a convertirse en un un grupo general concurrente nuevo (GGCN), un GGp también puede ser un grupo general concurrente (GGC) que está contenido en un GGnC.

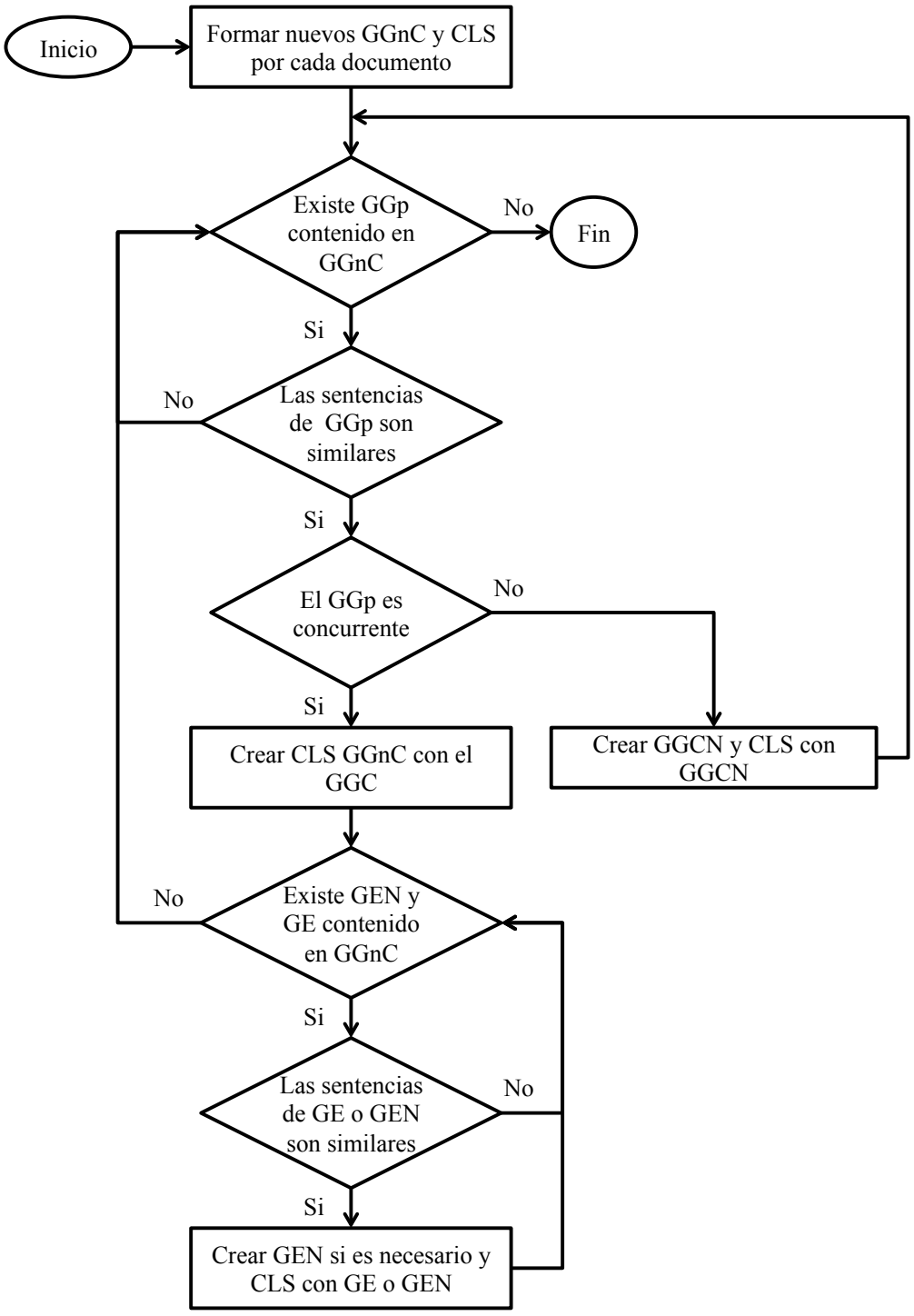

Figura 3.10: Diagrama general del poblado automático de la ontología

La creación de un GGCN o el hecho de que un documento está relacionado con un GGC es aprobado cuando las oraciones de un documento donde se encuentra una misma palabra clave son similares, lo cual se realiza mendiante la comparación de árboles sintácticos de oraciones descrita en la sección anterior. Una vez que se encuentra las oraciones son similares, si el GGp es un grupo no concurrente, es decir que se formó de la intersección de dos GGnC entonces se crea un GGCN así como las CLS correspondientes. Por otra parte, si el GGp es concurrente, es decir que se trata de un GGC contenido en el GGnC, se crea la CLS que relaciona el documento analizado con el GGC correspondiente, pero además se buscan saber si también existe relación con grupos específicos (GE) asociados al GGC o si es posible formar grupos específicos nuevos (GEN). De la misma forma que se forman grupos candidatos como GGp, en esta esta búsqueda se forman grupos específicos probables (GEp), cuyas oraciones asociadas a las palabras del GEp deben ser evaluadas por el método de similitud de oraciones para convertir un GEp en un GEN y crear la CLS entre un GE y el documento.

Existen otras consideraciones que se deben tomar en cuenta para hacer el poblado automático de la ontología correctamente. A continuación se describe con más detalle cada etapa del diagrama en la figura 3.10, además se explican algunas consultas que se deben hacer a la ontología para realizar algunas tareas del poblado automático, tal como la formación de nuevos grupos de palabras.

Descripción del método

En el inicio del poblado las primera tarea es crear los individuos autores ya que el documento necesita de los individuos autores para crearse. Una vez que los autores y documentos se crean se inicia la generación de nuevos grupos de palabras y las clasificaciones del documento. La primera etapa del poblado automático consiste en crear los GGp con sus características para poder ser tratados adecuadamente más adelante. En la figura 3.11 se muestran los pasos a seguir en la primera etapa del poblado. Una vez que los individuos de GGnC y las CLS del documento con el GGnC han sido creados inicia el procesamiento de cada GGnC en busca de GGCN o la relación con GGC y GE, así como la formación de un GEN. Algunos bloques de decisión en la figura 3.11 están marcados con un “*”, lo que indica que en ellos debe implementarse una consulta a la ontología para resolver la decisión. Para seleccionar un GGp el primer paso es seleccionar los grupos generales (GG) con la misma palabras clave que el GGnC y el siguiente paso es determinar si el GG es realmente un GGp.

Un GG es concurrente cuando su atributo *Concurred* es verdadero, lo cual significa que más de un documento está asociado a ese grupo de palabras. Se selecciona un GGp concurrente cuando un GG concurrente está contenido en el GGnC, es decir que la intersección entre GG y GGnC contiene los mismos elementos que GG. Cuando un GG es no concurrente, implica que se trata de un GG no relacionado con ningún otro documento, excepto el propio documento de donde surgió el grupo. Para formar un GGp a partir de un GG no concurrente, se debe tomar en cuenta si a partir del mismo GG no se ha formado un GGC, este aspecto es muy importante porque se trata de formar

grupos nuevos o de relacionar tantos documentos como sea posible por medio del grupo de palabras que coincide entre ellos, por lo tanto se debe dar la oportunidad a GGC de formar una relación con ellos, en lugar de formar nuevos GGC con las mismas palabras que otros. En la figura 3.12 se ilustra el conjunto de palabras que pueden formar un GGp, las cuales son apalabras que no se encuentran en otros GGC formados o relacionados con el mismo GG.

Cuando un GG es no concurrente se verifica si se han formado GGC a partir del GG y en caso de que sea así, el GGp se forma de la intersección del GGnC y la diferencia entre GG y los GGC, si esta intersección no es vacía se selecciona este conjunto y se nombra como un GGp no concurrente especial. En el caso en que no haya GGC creados con el GG la intersección entre el GGnC y el GG forma el GGp, que en caso de no ser un conjunto vacío, se selecciona el conjunto y se nombra como un GGp no concurrente.

En la figura 3.13 se muestra la segunda etapa del poblado automático, que es determinar si cada GGp puede convertirse en un GGCN o si el documento tiene relación con un GGC. En la selección de un GGp se verifica únicamente que el conjunto no sea vacío, sin embargo, en el siguiente paso se deben obtener las palabras contenidas en GGp, las cuales puede ser obtenidas mediante la intersección entre GGnC y GG cuando el GGp es diferente a un GGp no concurrente especial, y en caso contrario las palabras deben obtenerse de la intersección entre el GGnC y la diferencia entre GG y las palabras en los grupos GGC formados a partir del GG.

El siguiente paso es buscar similitud entre las oraciones de los documentos asociados a GGp y GGnC. Para obtener las oraciones en las cuales se puede buscar similitud se debe realizar una consulta a la base de datos. Esta tarea esta marcada en el diagrama de la figura 3.13 con “**”, mientras que las consultas a la base de datos están marcadas con “*”. Con las sentencias resultado de la consulta se utiliza el método para determinar la similitud de oraciones y si la similitud sobrepasa cierto umbral se determina que el GGp definitivamente tiene relación con el documento.

Las consecuencias de relacionar el GGp con el documento son descritas en las siguientes etapas del poblado automático. Una vez que las sentencias en todos los documentos han sido analizados, el proceso puede continuar con la búsqueda de relaciones con GE si las condiciones necesarias se cumplen, lo cual será descrito más adelante. En caso de que estas condiciones no se cumplan, el proceso continua con el análisis de el siguiente GGp seleccionado.

La etapa del poblado automático que continúa, lo hace a partir del conector marcado con el número dos en la figura 3.13 y se ilustra en el diagrama de la figura 3.14. Dependiendo de qué tipo de GGp sea el que aprobó la similitud de sus oraciones se pueden escoger dos opciones. En el caso de que el GGp no sea concurrente, el GGp se utiliza para crear el individuo de un GGCN así como las CLS consecuencia de haber encontrado la relación con el GGp, las cuales involucran al GGCN y al propio documento que está en proceso de análisis, así como el documento asociado al GGp, el cual también es un GGnC, que significa que tiene sólo un documento asociado a él. Con la creación del grupo nuevo y las clasificaciones, se puede continuar con el análisis de otro GGp. El GGp también puede ser un GGC, en este caso se tratará de encontrar GEN y relaciones nuevas con GE del GGC.

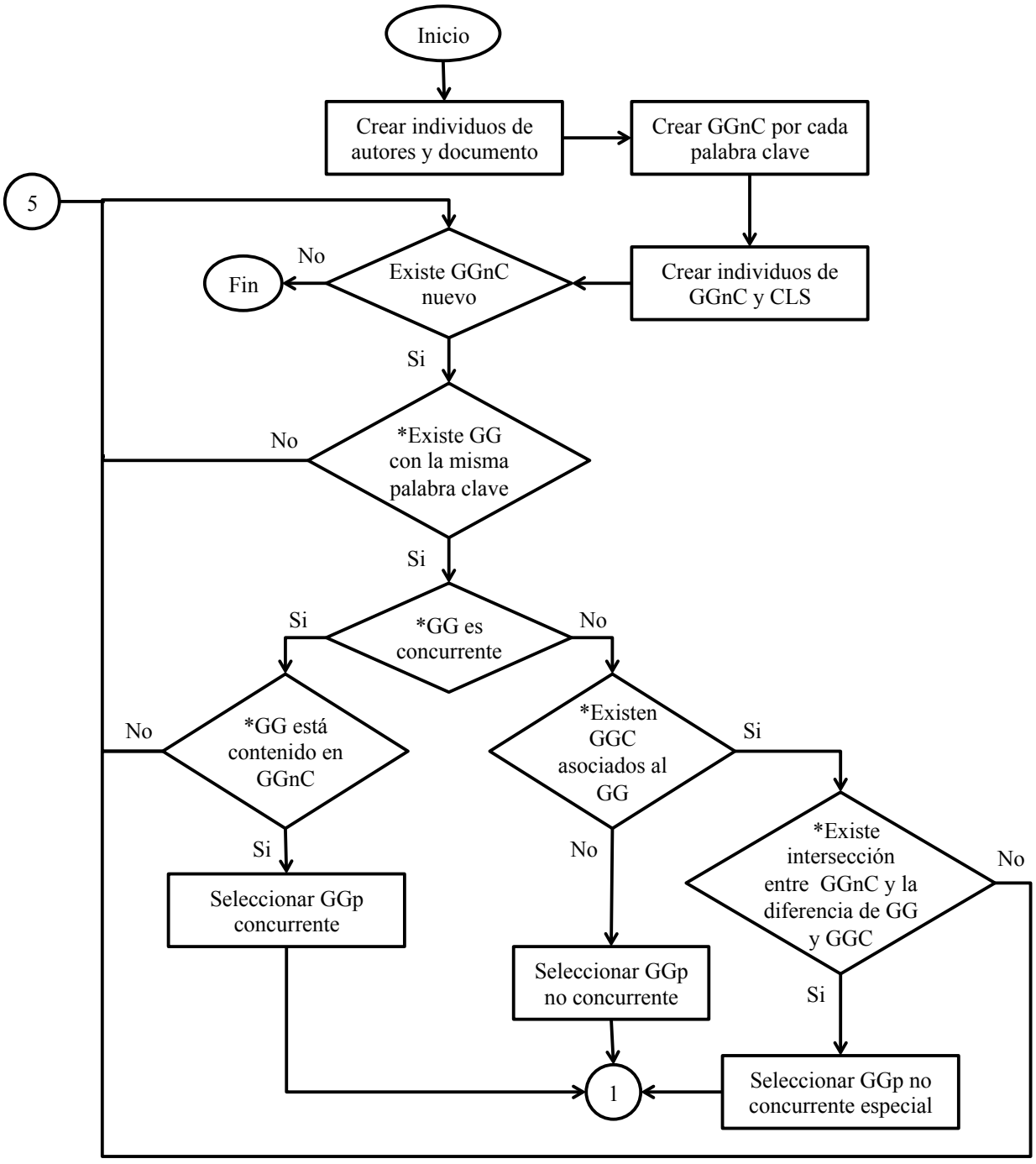

Figura 3.11: Selección de grupos generales probables contenidos en grupos nuevos

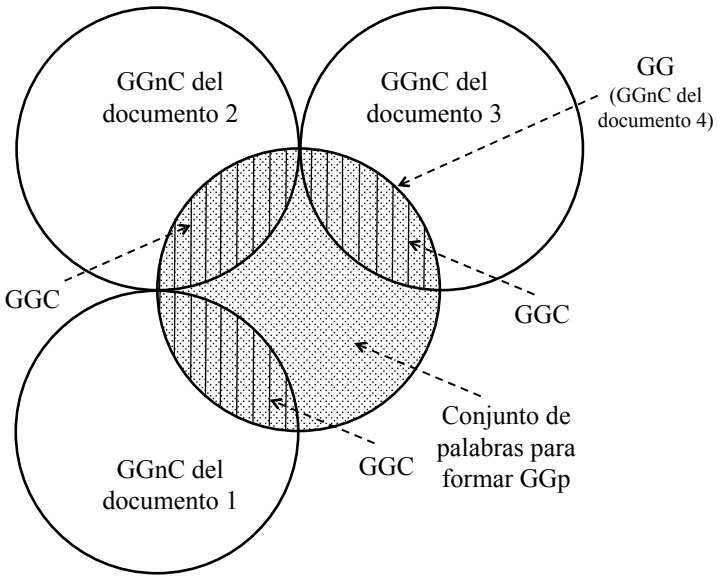

Figura 3.12: Conjunto de palabras para formar un nuevo grupo general probable

En primer lugar se busca formar un GEN, para lo cual se hace una nueva consulta a la ontología. Un GEN se forma de las palabras de un documento, por lo que el primer paso es obtener las palabras del GGnC que están en el documento del GGp, las cuales pueden obtenerse mediante una intersección, esto con el fin de encontrar más palabras que puedan coincidir entre el GGnC y el documento que se asoció al GGC. De este conjunto de palabras se puede obtener un GEN, sin embargo, se deben eliminar mediante diferencias, las palabras del propio GGC y sus GE asociados. Si el conjunto resultado aun contiene palabras, este conjunto es un GENp ya que para convertirse en un GEN aun debe pasar la prueba para encontrar similitud entre oraciones que contienen las palabras del GENp. La última prueba se realiza nuevamente con el método para encontrar similitud entre oraciones. Si se encuentra que sí existe una similitud entre las oraciones, se crea el GEN y las CLS que relacionen el documento que se está procesando con el GEN, así como la que relacione el documento del GGC con el GEN.

Cuando se finalice la búsqueda de GEN entonces se inicia la búsqueda de relaciones con GE del GGC, este proceso inicia en el conector marcado con el número tres en la figura 3.14. El conector tres lleva de nuevamente al diagrama de la figura 3.13, en la cual puede continuase con el análisis de más documentos que pueden llevar a la formación de otros GEN. Sin embargo, cuando los documentos terminen de analizarse y haya existido un documento que aprobó la similitud de oraciones con un documento que provino de un GGC, existen las condiciones para buscar relación con algún GE. Este proceso inicia en el conector marcado con el número cuatro en la figura 3.13, esta etapa es la última en el proceso del poblado automático de la ontología y sus pasos son descritos en la figura 3.15.

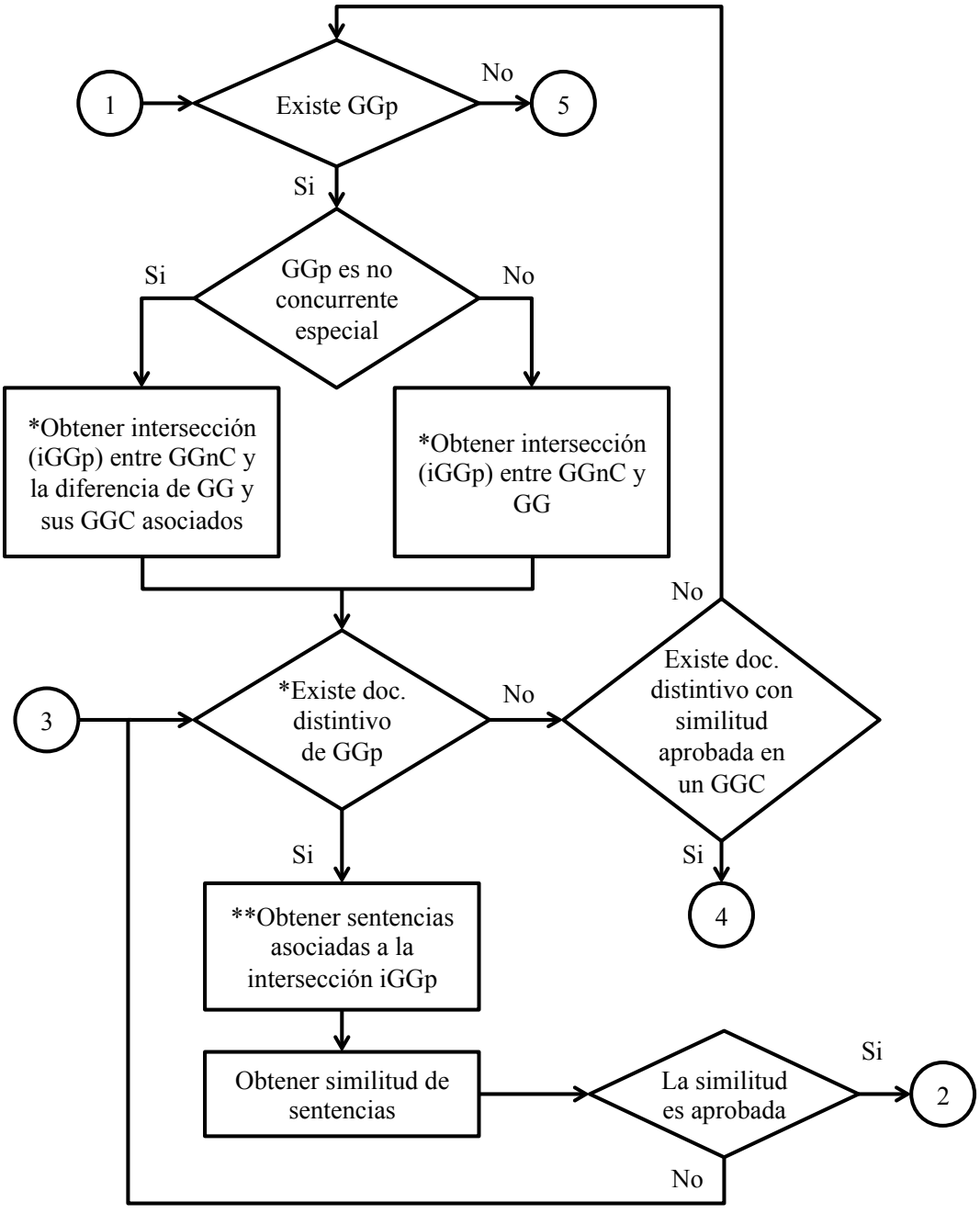

Figura 3.13: Aprobación de grupos generales mediante la comparación de sentencias

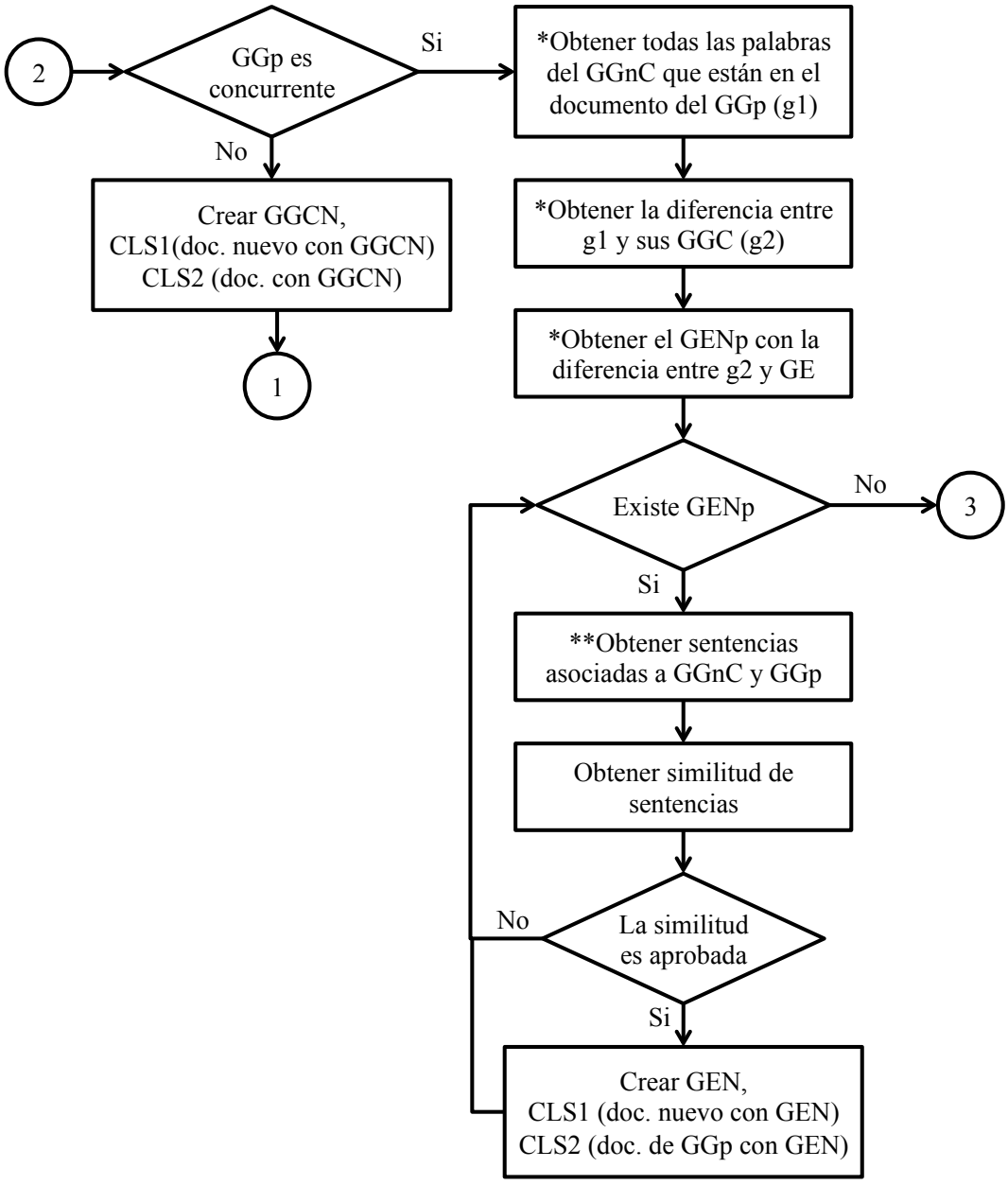

Figura 3.14: Creación de grupos generales nuevos y búsqueda de grupos generales específicos nuevos

El primer paso para encontrar relaciones con GE es hacer la consulta a la ontología para conocer si existen GE asociados al GGp. Cuando existe al menos un GE que pueda relacionarse, se verifica que el GE esté contenido en las palabras de GGnC y si es así el GE se convierte en un GEp para que sus oraciones sean analizadas con el método para encontrar similitud entre oraciones. Cuando

el GE pasa todas las pruebas para ser relacionado se crea la CLS entre el GE y el documento del GGnC. Al finalizar el análisis de los GE o cuando no haya existido ningún GE que se pueda relacionar, el análisis del GGp que es en este caso es un GGC, se termina haciendo la CLS entre el GGC y el documento que se está procesando. El análisis de un GGnC continua hasta terminar de procesar cada GGp obtenido. Finalmente cuando todos los GGnC, del documento nuevo que se analiza, han pasado por todo el proceso la clasificación del documento se da por terminada.

Con el poblado de la ontología, los individuos reflejan las relaciones que existen entre artículos científicos. Las clasificaciones (CLS) indican qué documentos están asociados a un grupo de palabras con un significado similar en las oraciones donde las palabras son utilizadas, así como qué grupos de palabras están contenidos en un documento. Los autores están relacionados con los grupos de palabras a través de los documentos en los que haya trabajado. De modo que, en la población de la ontología se encuentran las relaciones que existen entre documentos y los autores que se relacionan por medio de documentos similares en los que han trabajado. Para obtener la información sobre las relaciones descubiertas se deben implementar las consultas a la ontología.

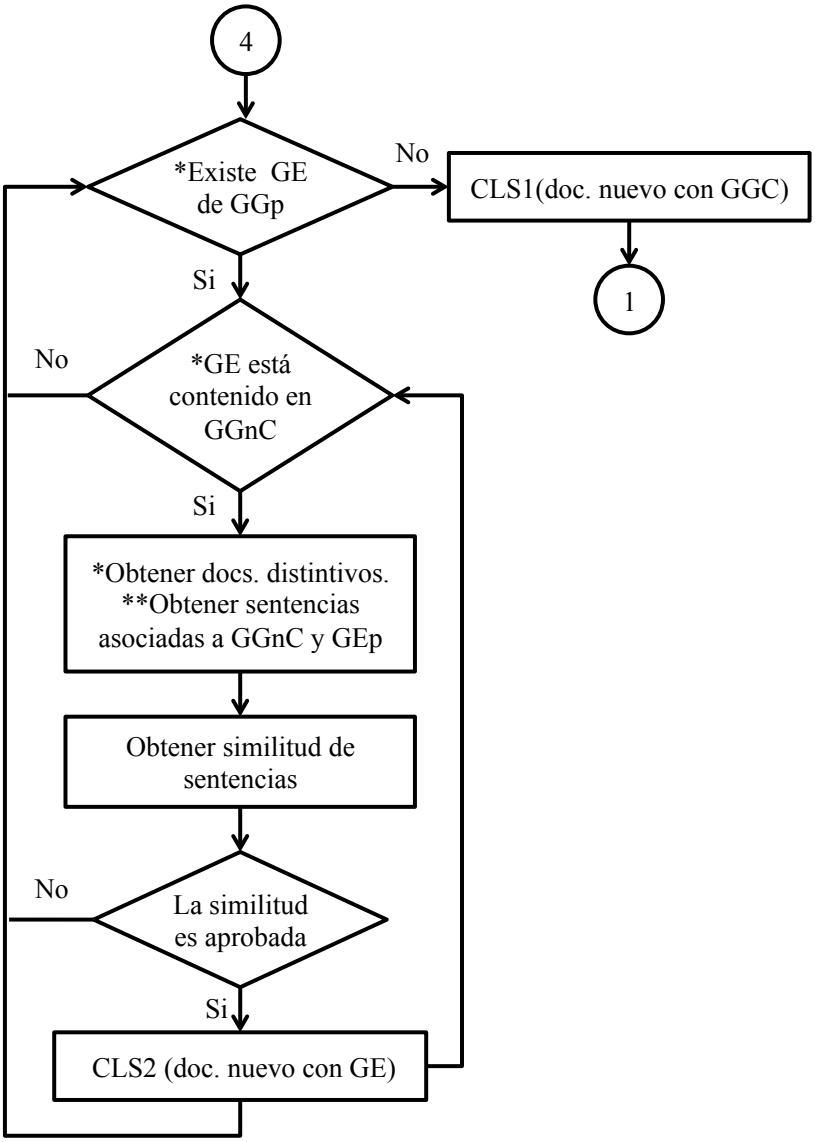

Figura 3.15: Selección y aprobación de grupos específicos

Capítulo 4

Implementación

Este capítulo describe la implementación del sistema propuesto en el capitulo anterior, que incluye el desarrollo de reglas en Jape para extraer los datos de artículos científicos y un sistema implementado en java para interactuar con GATE, procesar los datos extraídos de los documentos, hacer conexión con una base de datos y almacenar la información encontrada, así como manejar la ontología propuesta para el sistema y poblarla automáticamente por medio de consultas a la base de datos, la ontología y la comparación de oraciones, además buscar información de lo artículos y sus autores por medio de consultas a la ontología. El sistema consta de una interfaz gráfica para que el usuario pueda configurar el sistema, procesar artículos científicos y realizar las consultas.

En la primera sección del capítulo se describen las reglas implementadas en Jape, que son utilizadas por un módulo de GATE. Las siguientes secciones del capítulo describen la implementación del sistema por medio del desarrollo de clases en Java. La clase `Principal` es el núcleo del sistema, ya que en ella se encuentran los métodos que crean la interfaz gráfica del sistema, ejecutan todas las etapas que hacen el procesamiento de documentos y hacen las consultas para que el usuario pueda obtener información, por lo tanto esta clase está relacionada con muchas de la otras clases implementadas.

Las clases del sistema pueden clasificarse en varios grupos, las cuales corresponden a cada etapa en el procesamiento y consulta de los artículos. La segunda sección del capítulo explica el funcionamiento entre las clases que ejecutan los módulos de GATE y recuperan las anotaciones del los documentos. La tercer sección describe la conexión de la base de datos con el sistema y los métodos utilizados para manejar la ontología, introducir nuevos individuos en ella y hacer consultas. Cabe mencionar que los métodos para manejar la ontología y el API de GATE están basados en un proyecto que extrae datos de artículos científicos [26].

El método propuesto para hacer la comparación de oraciones es implementado mediante un conjunto de clases independientes que, por medio de la clase `RelacionesFrases`, interactúan con

la clase `Principal` durante el poblado de la ontología, la implementación del método es descrita en la cuarta sección del capítulo. Finalmente en las dos últimas secciones del capítulo se explica el poblado automático de la ontología y se presentan los métodos para crear la interfaz gráfica del sistema.

4.1. Reglas en Jape

Jape es el lenguaje en el que se implementan las reglas que JAPE utiliza para hacer el análisis sintáctico de superficie. Estas reglas se escriben en archivos de tipo jape, y son ejecutados de forma serial, junto con otros archivos similares que contienen reglas para identificar anotaciones más básicas, como lo son espacios, palabras, símbolos, sentencias y otros. Los archivos que contiene las reglas implementadas para identificar los datos que requiere el sistema son llamados después de todos los otros archivos que contiene las reglas básicas, y los mismos archivos que contienen las nuevas reglas son ejecutados en un orden específico ya que algunos de ellos requieren de otros, por ejemplo la regla para encontrar autor requiere que se ejecuten primero las reglas de "correo electrónico" y "universidad".

El orden descendiente en el que se ejecutan las nuevas reglas implementadas es: "correo electrónico", "universidad", "título", "autores", "sección", "palabras clave" y 'resumen". Estas reglas reconocen una serie de caracteres, identificando el dato buscado por la regla. Un archivo puede contener varias reglas que reconocen el mismo dato, pero estas reglas en el mismo archivo deben tener una prioridad para ejecutarse. A continuación se describen los patrones que cada regla busca para identificar cada uno de los datos que se buscan en el artículo.

- La regla para encontrar un **correo electrónico** se puede describir como: la aparición de una o más palabras, números, signos de puntuación, guiones bajos o puntos, seguidos por arroba, seguidos por una o más palabras, números, guiones bajos o signos de puntuación, opcionalmente seguidos por un punto, seguidos nuevamente por una o más palabras, números, signos de puntuación, guiones bajos o puntos, opcionalmente seguidos por una o más palabras o números.
- La regla principal para encontrar **instituciones** o universidades se puede describir como: una palabra característica en las organizaciones seguida por una palabra "of", seguida o no de una coma, seguida de al menos una serie de palabras que también sean un sustantivo y su primera letra sea mayúscula, seguidas de comas o apóstrofes.
- La regla para encontrar **autores** se describe como: un salto de línea seguido o no de espacios, seguido de un pronombre o palabra diferente de un verbo, conjugado en cualquier tiempo gramatical y diferente de una palabra escrita en minúsculas, seguida opcionalmente de un símbolo, palabra o signo de puntuación diferente a coma (esto identifica el primer nombre del autor). Esto seguido de un patrón que puede repetirse hasta tres veces y se compone

de un espacio seguido de una palabra o pronombre diferente a una palabra en minúsculas, opcionalmente seguida de un símbolo palabra o signo de puntuación, otra opción que puede aparecer en lugar de la palabra o pronombre mencionados son las palabras "de", "van" o variables similares, (este patrón obtiene el segundo, tercer o cuarto nombre de autor). Esta serie de palabras, símbolos y signos de puntuación que componen el nombre de un autor puede estar seguida de una coma o la palabra "and" seguido de una coma o seguido de una anotación de tipo "correo electrónico" o de "universidad".

- La regla para obtener el **título** se describe como: cero o más saltos de línea, seguidos por una o más series de palabras y espacios, seguidos de al menos un salto de línea, que se encuentren en máximo dos renglones.
- La regla una **sección** se puede describir como: la aparición opcional del número uno seguido o no de un punto y un espacio, seguida de la palabra "Introduction", "References", "Background" o un símbolo, seguido de un salto de línea.
- La regla principal para obtener el inicio del **resumen** se describe como: la palabra "Abstract"que inicie con mayúscula, seguida de cero o más signos de puntuación, guiones o símbolos, seguidos de cero o más un sustantivos que inicien en mayúscula, seguido de uno o más espacios.
- La regla para encontrar las **palabras clave**, puede describirse como: la aparición de la cadena "Index Terms" o "KeyWords", seguidas de cero o más símbolos, espacios o saltos de línea, seguidas de máximo cuatro series que se componen de cero o más palabras, espacios, signos de puntuación o números, que sean diferentes de una anotación de tipo "sección" y estén seguidas de un salto de línea.

Estas reglas son utilizadas por uno de los módulos de GATE que son ejecutados por instrucciones de uno de los métodos en la clase `Principal` del proyecto en java. Las siguientes secciones describen el procesamiento de artículos científicos que inician con el anotado de los artículos mediante las reglas que se han mencionado.

4.2. Uso de GATE con Java

La primera etapa en el procesamiento de cada artículo científico es la extracción de datos, lo cual se realiza mediante el API de GATE que permite ejecutar módulos de ANNIE para etiquetar los datos que se buscan en el documento. La clase `Principal` contiene el método para iniciar este proceso. La figura 4.1 muestra el conjunto de clases relacionadas con la clase `Principal` que utilizan el API de GATE para ejecutar los módulos necesarios y extraer los resultados del proceso que hace el anotado de documentos.

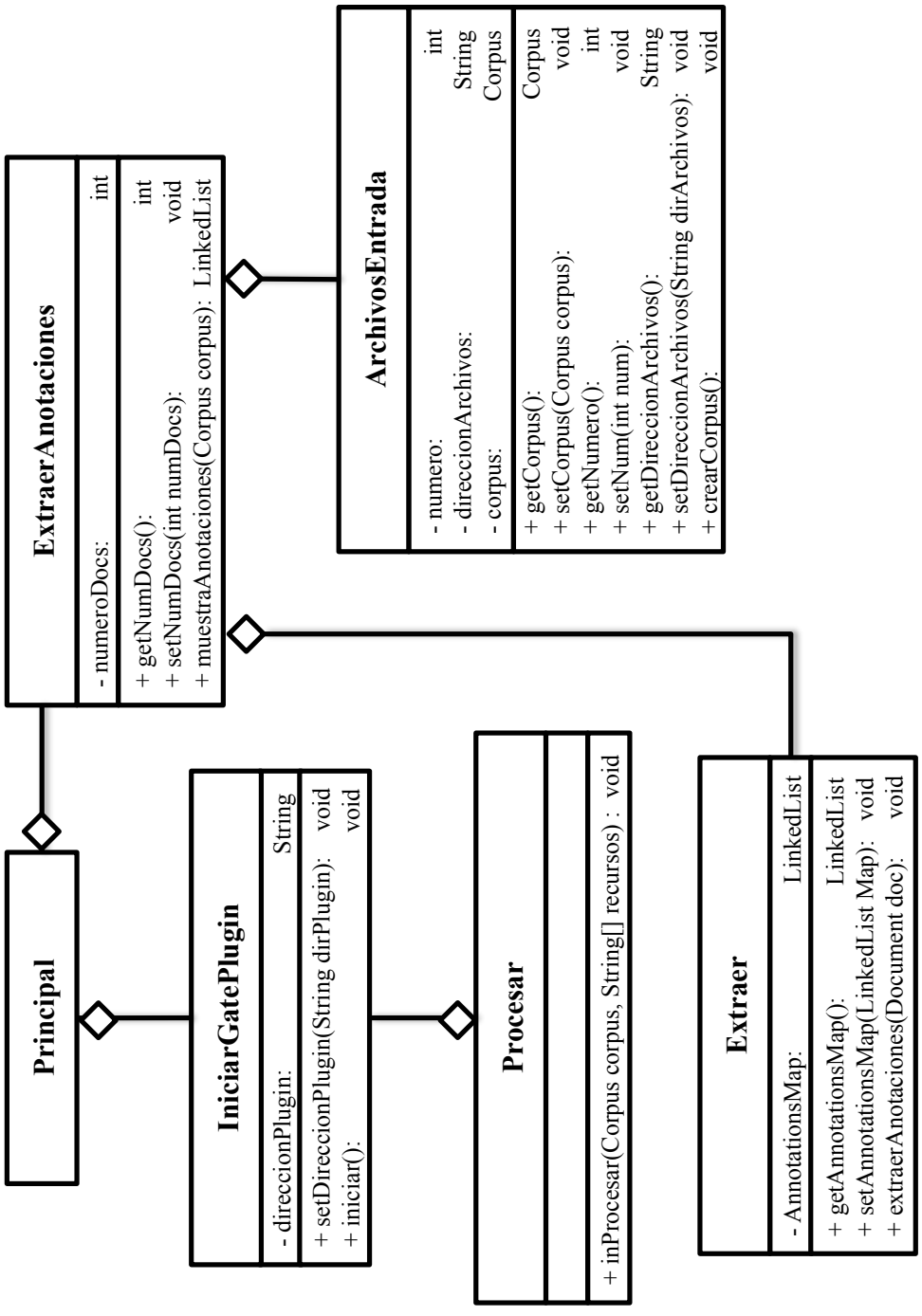

Figura 4.1: Diagrama de clases con la clase Principal y otras clases para interactuar con GATE.

La primer clase necesaria para ejecutar los módulos de GATE es `iniciarGatePlugin`, a través de ella se accede a los recursos de GATE, para ejecutar los módulos de GATE se utiliza la clase `Procesar` en donde se indican cuáles y en qué orden se ejecutan los módulos de GATE. Estos módulos son:

- AnnotationDeletePR: limpia el documento, haciendo posible leer todos los caracteres del documento.
- DefaultTokeniser: identifica los caracteres del documento.
- SentenceSplitter: divide el documento en sentencias.
- POSTagger: realiza el etiquetado gramatical de cada palabra, es decir asigna la categoría gramatical a cada palabra.
- DefaultGazetteer: identifica entidades nombradas.
- ANNIETransducer: identifica los sintagmas nominales y verbales del documento mediante reglas.

Los parámetros que necesita el objeto `Procesar` para funcionar son la cadena de caracteres para indicar los recursos de GATE que se deben ejecutarse y el corpus. El conjunto de documentos que se desea procesar forman el corpus, el cual puede ser manipulado mediante la clase `ArchivosEntrada`. Una vez que se realiza el procesamiento de los documentos en el corpus, lo siguiente es utilizar la clase `ExtraerAnotaciones` con la cual mediante el objeto `ArchivosEntrada` y la clase `Extraer` se crea una lista con las anotaciones de todo el documento, listas para ser procesadas nuevamente y obtener los datos finales que son útiles para las siguientes etapas del procesamiento. A continuación se hace una breve descripción de estas clases.

- **IniciarGatePlugin**: Utiliza un API para cargar los recursos de GATE, que se encuentran previamente instalados, el API únicamente necesita tener la dirección de los recursos.
- **Procesar**: Los recursos que se utilizan en el procesamiento son módulos de ANNIE por eso, mediante el API que provee GATE, este módulo es activado para poder cargar los recursos que se requiere, estos son ejecutados sobre el corpus indicado.
- **ArchivosEntrada**: esta clase es utilizada para tener control sobre el corpus que maneja GATE, ya que los atributos de éste tal como el número de documentos que almacena y el propio corpus son útiles no sólo para iniciar el anotado, sino también para facilitar la extracción de los datos anotados o etiquetados en los documentos.
- **ExtraerAnotaciones**: se utiliza para obtener las anotaciones de cada documento a través del corpus, creando la lista de anotaciones mediante la clase `extraer`.

- **Extraer**: durante el anotado de cada documento, GATE identifica una serie de anotaciones que no son de interés para el sistema, tales como espacios, caracteres y otros, además de anotaciones que pueden parecer útiles pero no lo son, tales como nombres propios que pueden parecer nombres de autores, esta clase extrae las anotaciones que sí son útiles para el sistema. Dependiendo del modo de control con que se ejecute una regla, ésta genera una o muchas anotaciones. Un ejemplo es la regla para etiquetar el título del artículo, que genera una sola anotación, sin embargo, las otras reglas son capaces de generar cualquier cantidad de anotaciones. De las anotaciones que se generan es probable que no todas correspondan a la información que se pretende obtener. En el método principal de esta clase se recorren dos veces las anotaciones que se encuentran en el documento. En el primer recorrido se trata de identificar el inicio del resumen, la primera sección y las palabras clave, se determina en qué orden de aparición de encuentran, si es que fueron encontradas y se obtiene la anotación que marca el limite del documento donde de encuentran las anotaciones útiles. En el segundo recorrido de las anotaciones se determina el final del resumen, el final de las palabras claves si es que no estaba muy claro y se filtran las anotaciones que formaran la lista de anotaciones que entrega la clase `ExtraerAnotaciones`.

4.2.1. Procesamiento de etiquetas de GATE

Algunos datos que se recuperan con la extracción de datos con GATE tales como título o institución, pueden ser almacenados en la base de datos y la ontología inmediatamente, sin embargo, los autores, correos electrónicos, palabras clave y resumen deben ser procesados nuevamente para mejorar la extracción de los datos. La figura 4.2 muestra las clases que se utilizan para realizar este procesamiento y a continuación se describen brevemente sus funciones.

- **NormalizaAutor**: la regla que etiqueta los autores, marca el conjunto de nombres de autores, o el nombre de los autores seguido de correos electrónicos o instituciones, esta clase tiene como función obtener la lista real de autores del artículo. Este método separa la cadena etiquetada por GATE, identifica qué partes de la cadena son correos electrónicos o instituciones y las elimina.
- **NormalizaEmail**: el correo electrónico generalmente no tiene que ser procesado por segunda ocasión, sin embargo, en algunos casos, por ejemplo correos electrónicos que pertenecen al mismo dominio, se encuentran en un formato diferente que debe separarse para que puedan ser asignados a su propio autor. Con esta clase el correo electrónico es separado para crear el correo electrónico en el formato palaba@dominio.
- **NormalizaKeyWord**: la serie de palabras clave que se identifican con GATE son uno o varios renglones, que contienen las palabras clave separadas por algún signo de puntuación. En esta clase se elimina la palabra que marca el inicio de la lista de palabras clave y separa las palabras clave para poder ser almacenadas por separado y no como un texto.

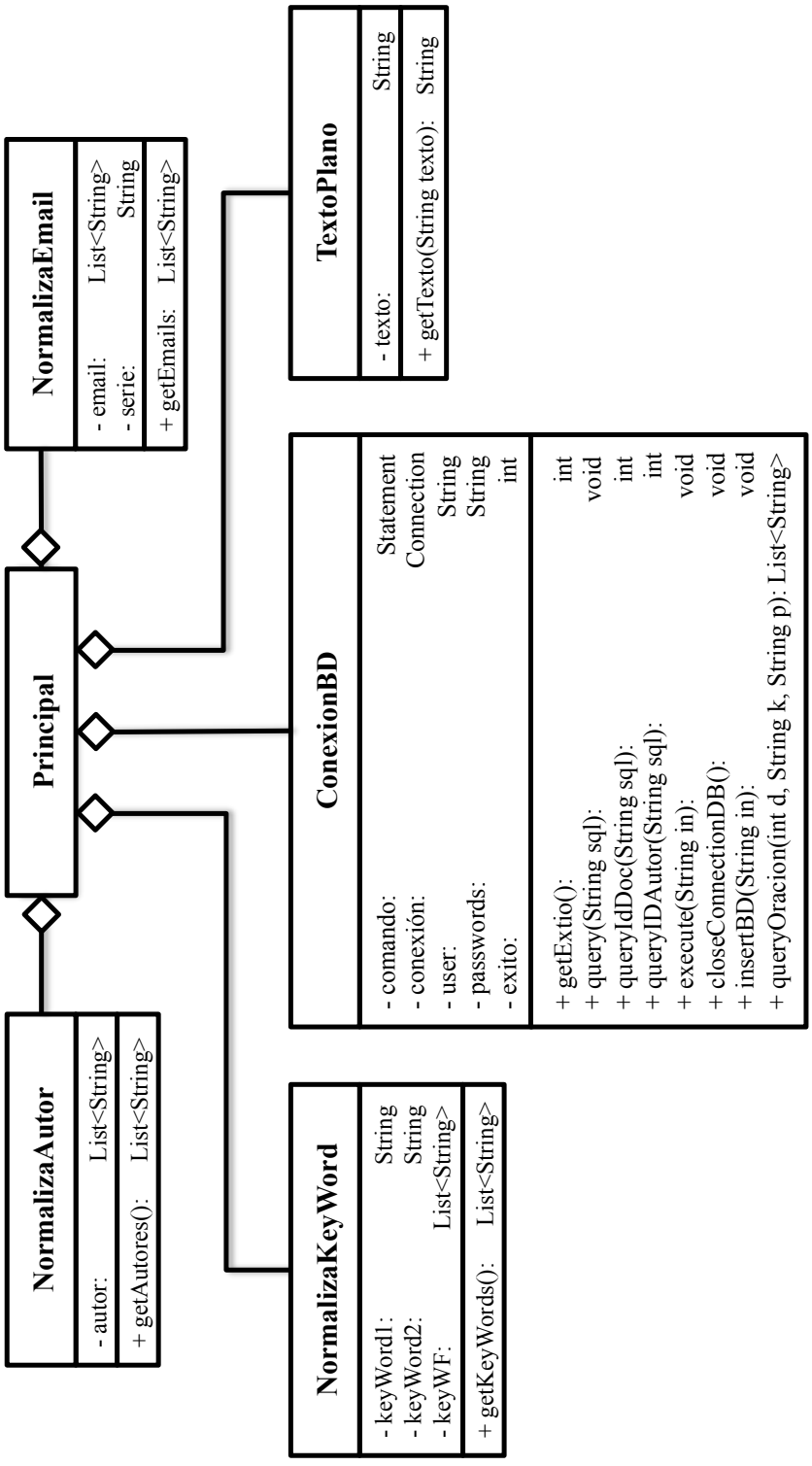

Figura 4.2: Diagrama de clases con la clase Principal y otras clases para procesar los datos etiquetados por GATE.

- **TextoPlano**: es común que el resumen se presente en la parte del artículo que viene en dos columnas, lo que provoca que haya muchas palabras separadas por un guión, que indica que la otra parte de la palabra está en el siguiente renglón. Esta clase tiene como función eliminar la palabra que indica el inicio del resumen y eliminar los guiones de las palabras para que al procesar las oraciones, éstas puedan ser interpretadas correctamente.

4.3. Almacenamiento de datos

Esta sección presenta la implementación de los métodos que se utilizan para almacenar y manejar los datos extraídos de los documentos. Todos los datos extraídos son almacenados en la base de datos, mientras que en la ontología se almacenan sólo los datos que ayudan a representar relaciones entre autores y documentos.

4.3.1. Conexión a la base de datos

Una vez que la extracción de datos ha finalizado, éstos deben ser almacenados en la base de datos, posteriormente éstos mismos serán almacenados en la ontología; el poblado de la ontología se hace en otra etapa del procesamiento. La figura 4.2 muestra también la clase `ConexionBD` la cual tiene como función hacer la conexión con el manejador de base de datos postgres, sobre el cual se ha implementado la base de datos que se propuso, de acuerdo al diagrama entidad-relación (figura 3.3) que se presentó en el tercer capítulo. A continuación se describen otras funciones que incluye esta clase.

- **ConexionBD**: esta clase contiene los métodos para hacer la conexión a postgres por medio de un usuario y una contraseña, además implementa los métodos para insertar autores, documentos, oraciones y relaciones entre autores y documentos. Entre las consultas que se hacen a la base de datos, existen algunas que se hacen comúnmente para hacer el poblado de la ontología, por lo que se implementan los métodos para obtener el identificador de un autor y un documento, así como para obtener la lista de palabras relacionadas a una palabra clave en una sola oración, esto es útil para obtener las oraciones que cumplen los requerimientos para ser comparadas.

4.3.2. Manejo de la ontología

El manejo de la ontología permite cargar las clases de la ontología, propiedades de los objetos individuos e importar las ontologías que la propia ontología necesite, además de permitir hacer cambios o consultas a la ontología.

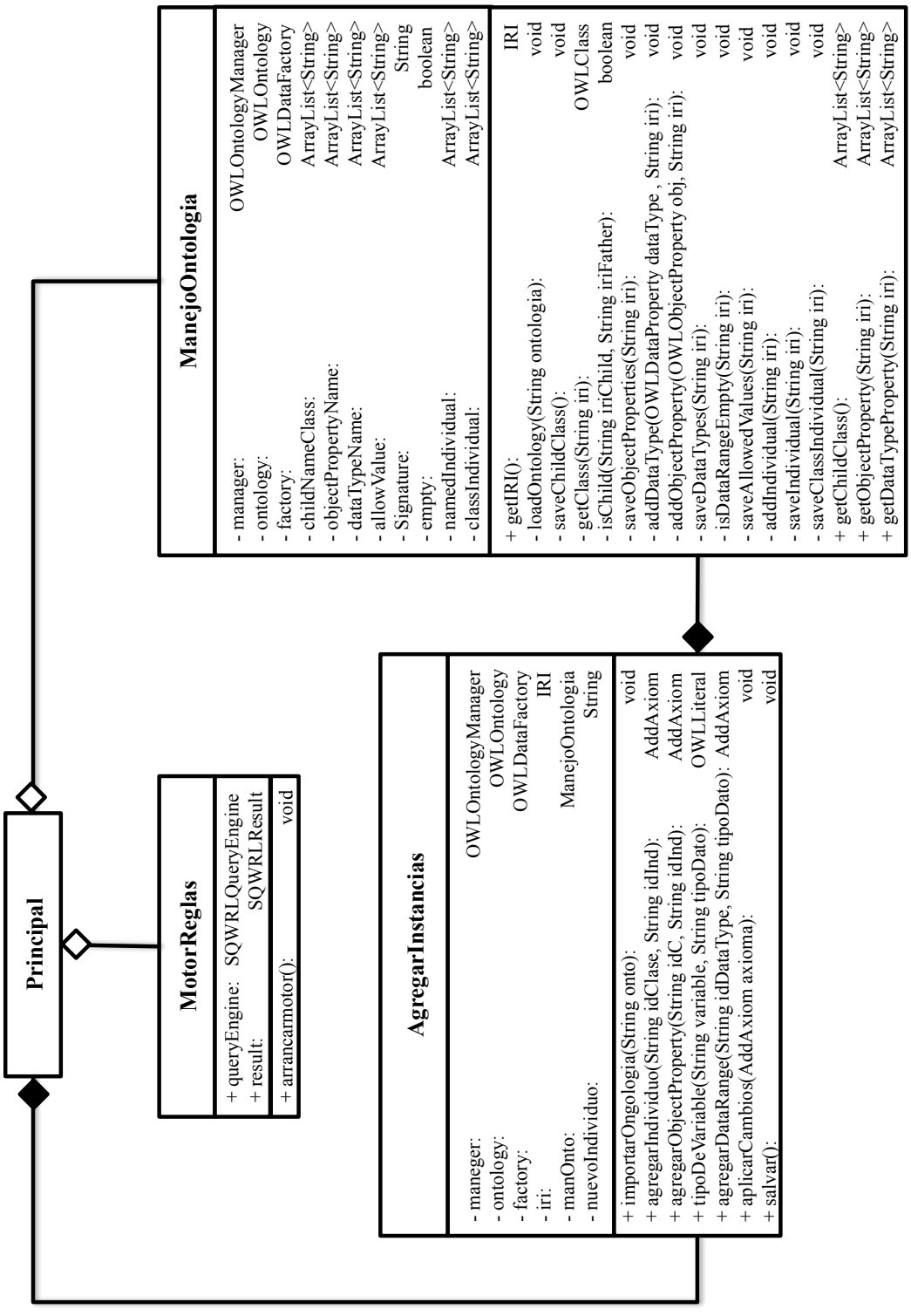

Figura 4.3: Diagrama de clases con la clase Principal y otras clases para manejar la ontología.

La figura 4.3 presenta las tres clases para manejar la ontología, la clase `Principal` interactúa principalmente con la clase `AgregarInstancias` que sirve para facilitar el poblado de la ontología al momento de agregar individuos, mientras que la clase `ManejoOntologia` contiene los métodos que se requieren para cargar la ontología correctamente y asegurar que no existan errores para leerla y hacer cambios en ella. A continuación se describen brevemente las funciones de estas clases:

- **ManejoOntologia**: esta clase implementa los métodos para cargar la ontología y leer las clases y subclases de la misma, verifica que las clases pertenezcan a un dominio, guarda las propiedades de objeto y de clase, guarda los tipos de datos que pertenecen a una clase, verifica que los tipos de datos o propiedades de objeto no tengan valores predefinidos para evitar asignar valores incorrectos en un individuo, o bien guarda los valores permitidos en un rango de datos, también guarda los individuos de una clase, puede identificar las clases que no tiene subclases.

- **AgregarInstancias**: la función principal de esta clase es facilitar la forma de insertar nuevos individuos en la ontología. Esta clase carga la ontología por medio de un objeto de la clase `ManejoOntologia` y utiliza sus métodos para agregar individuos utilizando como parámetros únicamente el nombre de la clase y el nombre que se le quiere dar al individuo, también agrega propiedades de objeto indicando únicamente el dato que se quiere agregar y nombre de la propiedad donde se agrega, lo mismo sucede para agregar tipos de datos.

- **MotorReglas**: contiene los pasos necesarios para ejecutar el motor de consultas SQWRL para hacer las consultas al sistema, sin tener que leer la ontología completa.

4.4. Implementación del método para comparación de oraciones

Los datos que se extraen de los documentos se almacenan en la base de datos y la ontología. Para insertar datos en la base de datos sólo se necesita saber qué tipo de dato es el que se inserta y si pertenece a autor o a un documento, sin embargo, para poblar la ontología es necesario tomar en cuenta otras consideraciones, entre ellas es necesario saber la relación entre las oraciones del documento que se intenta clasificar con otros documentos procesados anteriormente por el sistema. En esta sección se describe la implementación del método propuesto para hacer la comparación de oraciones.

El objetivo del método es encontrar un grado de similitud entre un par de oraciones. Para comparar dos oraciones, es requisito que ambas compartan al menos dos palabras similares, es decir que las palabras sean iguales después de eliminar su variedad morfológica. Si las oraciones cumplen los requisitos para ser comparadas se obtiene el árbol sintáctico de las oraciones y por cada dos palabras similares que compartan se obtiene el subárbol que contiene las dos palabras similares, estos subárboles se recorren en busca de nodos similares y se calcula el grado de similitud.

Finalmente se calcula el promedio de grados de similitud de los subárboles comparados y se obtiene el grado de similitud final de la oración.

La figura 4.4 muestra el diagrama de clases de la implementación del método. La clase `Relacion Frases` ejecuta la serie de pasos para obtener la relación entre cada par de oraciones que se deseen comparar, mediante la clase `Similitud` obtiene el grado de similitud entre cada árbol sintáctico comparado, los cuales son obtenidos a través de la clase `ArbolSintactico`. Para identificar el par de palabras que las oraciones comparten, si es que existe, se utiliza la clase `CoincidenciasPalabras`, la cual retorna objetos de tipo `NodoClase` cuya función es caracterizar la palabra que comparten dos oraciones, es decir identifica la palabra y el nodo de cada árbol donde ésta aparece. Los subárboles que se comparan pueden formarse con la clase `SubArbol` a partir del árbol sintáctico de la oración y los nodos de las palabras compartidas en las oraciones. La clase `ComparaArboles` hace el recorrido de los subárboles para identificar nodos similares para que un objeto de la clase `Similitud` pueda calcular el grado de similitud entre las oraciones. A continuación se describen estas clases.

- **RelacionFrases**: esta clase implementa los métodos necesarios para obtener el grado de similitud de un par de oraciones, creando los objetos que servirán de parámetros para que otras clases puedan funcionar, asegurando que las oraciones cumplan los requisitos necesarios para ser comparadas y creando los árboles sintácticos que sean necesarios para entregar un grado de similitud entre un par o más de oraciones. Así como realizar el cálculo del grado de similitud entre subárboles de cada par de oraciones.

- **ArbolSintactico**: los métodos de esta clase son utilizados para crear el árbol sintáctico de una oración, por medio del API de *Stanford Parser*. Además, durante la creación del árbol es posible eliminar la variedad morfológica de cada palabra e identificar las palabras importantes a partir de las cuales se pueden formar subárboles que pueden ser recorridos y comparados. El reconocimiento de estas palabras también es útil para insertar información sobre las oraciones en la base de datos, ya que con consultas posteriores a la base de datos se puede identificar fácilmente cuáles son las oraciones que pueden ser candidatas a someterse a una comparación con oraciones de un documento nuevo.

- **CoincidenciasPalabras**: los métodos de esta clase forman la lista de pares de palabras similares que comparten dos oraciones. Esta lista surge de la combinación de palabras similares en cada oración, de modo que si dos oraciones comparten tres palabras similares (p_1, p_2, p_3) la lista se forma con tres pares de palabras ((p_1, p_2),(p_1, p_3),(p_2, p_3)), que servirán para tres comparaciones entre los subárboles que las contengan.

- **NodoFrase**: un objeto de esta clase contiene la palabra similar que comparten dos oraciones, también contiene el número de nodo donde aparece la palabra en cada árbol de las oraciones que se comparan, esta clase es muy útil porque permite tener una referencia de la ubicación de la palabra y el inicio de los recorridos del árbol, a pesar de tomar diferentes subárboles para recorrer y comparar.

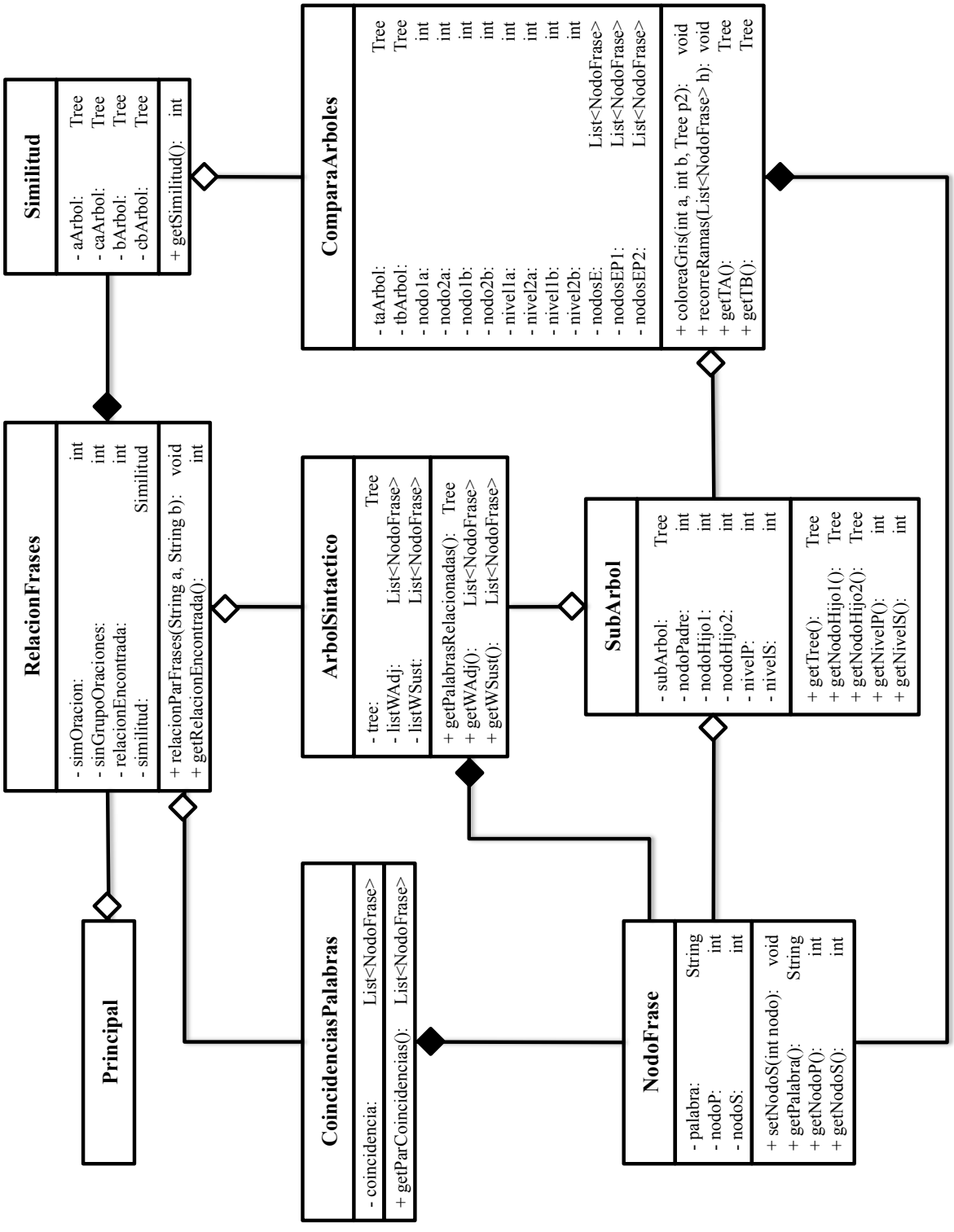

Figura 4.4: Diagrama de clases con la clase principal y otras clases que realizan la comparación de oraciones.

- **SubArbol**: los atributos de un subárbol de un árbol sintáctico son el propio subárbol que contiene las palabras similares, los nodos hijo del nodo raíz que siguen el camino hacia las palabras similares y los niveles de dichos nodos hijo. Estos atributos son útiles para iniciar los recorridos de los árboles, debido a que el recorrido en busca de nodos similares, inicia por estos caminos.

- **ComparaArboles**: esta clase realiza el recorrido de un par de árboles en busca de nodos similares para marcar, sus atributos son el propio par de árboles que compara y los nodos donde debe iniciar el recorrido, así como los niveles de los nodos en cada árbol que son comparados durante el recorrido y los nodos que son similares y dan inicio a nuevos recorridos hacías las ramas de los árboles, estos nodos necesitan ser identificados por número de nodo y el valor que contienen por lo que se instancias con objetos de tipo `NodoFrase`. Los métodos que implementa la clase se utilizan para colorear en las ramas de los árboles que no tienen coincidencia entre ellos y para hacer recorridos recursivos hacía las ramas de los árboles.

- **Similitud**: después de que los métodos de la clase `ComparaArboles` han marcado los nodos que son similares en los árboles comparados, esta clase hace el conteo de nodos marcados y asigna un valor a cada nodo según su valor gramatical, para calcular el grado de similitud entre los árboles.

4.5. Poblado de la ontología

La siguiente etapa durante el procesamiento de un artículo científico, después de extraer sus datos y almacenarlos en la base de datos, es crear individuos en la ontología que contengan datos del artículo y su relación con otros artículos previamente procesados. En el capítulo tres sección 3.3.2 se describe el procedimiento para crear estos individuos, que pueden ser de tipo autores, documentos, grupos generales, grupos específicos o clasificación. Los individuos de tipo autores y documentos se crean directamente de los datos que se extraen del documento, tal como se inserta información en la base de datos, sin embargo la creación de individuos de grupos y clasificación se realiza mediante un procedimiento orientado a consultas a la ontología y base de datos, así como el uso del método para comparar oraciones. Este procedimiento está implementado en la clase `Principal` en el método `iniciarButtonActionPerformed()` que comienza por hacer la extracción de los datos del artículo, inserta los datos en la base de datos y finalmente utiliza el procedimiento para crear los correspondientes individuos en la ontología.

Las consultas a la ontología que se hacen para crear nuevos individuos se realizan con el objetivo de encontrar grupos de palabras relacionadas con el artículo que se está procesando. Las consultas por lo tanto realizan uniones e intersecciones entre grupos de palabras de artículos que probablemente tengan similitud con el artículo que se procesa y el grupo de palabras del propio artículo que se procesa. Las consultas más relevantes que se usan para determinar la formación de nuevos individuos de grupos de palabras y clasificación son cinco y se mencionan a continuación.

1. Encontrar grupos generales asociados al artículo está determinado por: la existencia de grupos generales con la misma palabra clave que el grupo de palabras nuevo del artículo.

2. Encontrar relación con grupos generales concurrentes está determinado por: el grupo general está contenido en el grupo de palabras nuevo del artículo.

3. Encontrar grupos de palabras que aun no han sido asociados a ningún grupo general concurrente está determinado por: la existencia de grupos generales concurrentes asociados a un grupo general.

4. Encontrar relación con grupos de palabras que no han sido asociados a ningún grupos general concurrente está determinado por: el grupo de palabras del nuevo artículo está contenido en la diferencia de palabras de un grupo general y los grupos generales concurrentes asociados al grupo general

5. Encontrar probables grupos específicos nuevos está determinado por: la existencia de un grupo de palabras que estén en el documento asociado al grupo general y el grupo de palabras del artículo nuevo, pero que no estén en ningún grupo general concurrente asociado al documento y ningún grupo específico asociado al grupo general.

4.6. Interfaz de usuario

En esta sección se presenta la interfaz de usuario que se desarrolló con el objetivo de que el usuario pueda especificar la dirección de los recursos que necesita el sistema que estén previamente instalados y proporcionar el usuario y contraseña para acceder a la base de datos. Por medio de la interfaz el usuario puede indicar la carpeta que contiene los documentos que quieren analizarse y además proporciona una forma sencilla para hacer consultas al sistema y obtener información sobre artículos, autores y las relaciones que se descubrieron entre ellos.

La figura 4.5 muestra la implementación de la clase `Principal`, donde se implementa la interfaz de usuario, además del procesamiento de artículos que utiliza todas las clases que ya se han mencionado, la figura también muestra otras clases que se utilizan para configurar el sistema y manejar la información que viene de las consultas a la ontología para que pueda ser útil en la interfaz de usuario. A continuación se describen estas clases.

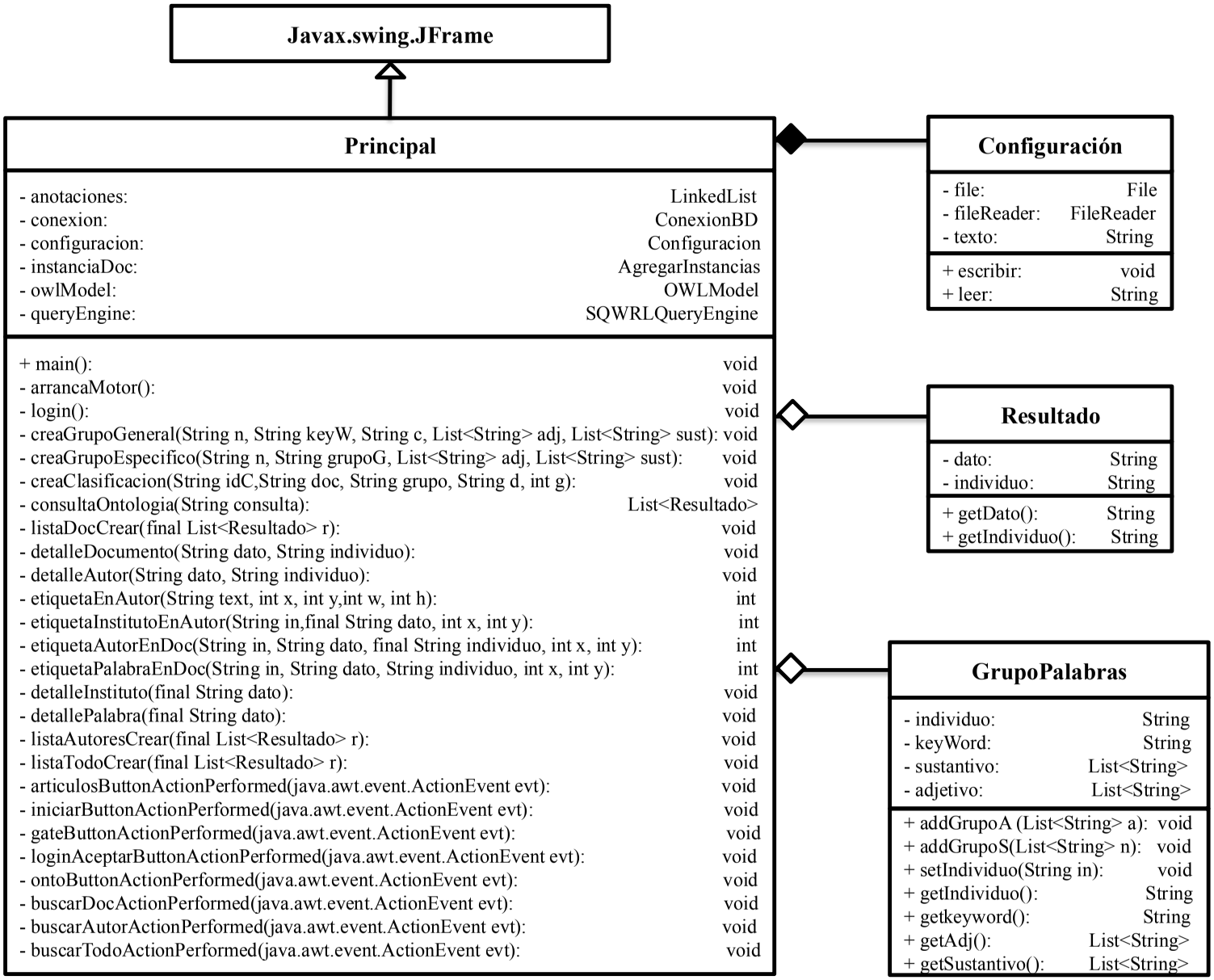

Figura 4.5: Diagrama de clases con la clase Principal y otras clases para interactuar con datos de la ontología

- **Principal**: esta clase hereda los métodos de la clase `Javax.swing.JFrame` para implementar la interfaz gráfica para el usuario. Por medio de la interfaz se puede iniciar el análisis de un artículo científico, por lo que en esta clase se implementa todo el procedimiento de extracción de datos, almacenamiento de datos y poblado de la ontología, para lo cual es necesario tener los métodos para acceder a la base de datos, cargar la ontología e iniciar el motor de consultas a la ontología y crear nuevos individuos. Otros métodos de esta clase están hechos para manejar las ventanas de la interfaz y crear vínculos que muestren mayor información sobre un dato específico. El resto de los métodos se utilizan para obtener datos que permitan el acceso a recursos de GATE, la ontología y la base de datos.
- **Configuracion**: los parámetros que pueden ser configurados por el usuario, son las rutas de los directorios donde se encuentran los recursos que el sistema requiere, éstos son la carpeta donde se instaló GATE y el archivo de la ontología, otros parámetros que el sistema necesita son el nombre de usuario y contraseña para acceder a la ontología. Estos parámetros se guardan en un archivo y pueden ser creados o modificados por medio de los métodos de esta clase.
- **GrupoPalabra**: para crear nuevos individuos que resultan de intersecciones entre conjuntos de palabras de otros individuos de la ontología, es necesario manejar los atributos de los nuevos individuos en un objeto, esta clase permite instanciar estos objetos y una vez que el nuevo individuo está modelado, insertarlo en la ontología.
- **Resultado**: en la interfaz de usuario se muestran varios vínculos que permiten obtener más información sobre un resultado de una consulta. Para obtener más información sobre un resultado se requieren varios parámetros, que ayuden a formar una nueva consulta, esta clase modela la información sobre el resultado de una consulta a la ontología, de modo que el resultado sea útil para consultar más información sobre él.

Para consultar información de la ontología se implementaron consultas que pueden ser ejecutadas por el usuario desde la interfaz gráfica, a continuación se mencionan estas consultas:

- Búsqueda de documentos por palabra en título.
- Búsqueda de documentos por palabra en palabra clave.
- Búsqueda de documentos por palabra en resumen.
- Búsqueda de documentos por palabra en todo.
- Búsqueda de documentos por autor.
- Búsqueda de documentos por institución.
- Búsqueda de autores relacionados a palabra clave.

- Búsqueda de autores relacionados a otro autor.
- Búsqueda de autores relacionados a una institución.
- Búsqueda de todos los documentos.
- Búsqueda de todos los autores.
- Búsqueda de todos las instituciones.
- Búsqueda de todas las palabras clave.
- Búsqueda de palabras clave relacionadas con otra palabra clave.

Las búsquedas de autores relacionados con otro autor, deben dar como resultado los autores que hayan trabajado con el autor que indique el usuario, en un mismo artículo. A diferencia de la búsqueda de autores relacionados con palabra clave, que dan como resultado autores que hayan escrito artículos que contengan la palabra clave que indique el usuario, independientemente de que hayan trabajado juntos en un mismo artículo o no. La última consulta sobre palabras clave relacionadas con otra palabra clave indicada por el usuario, da como resultado las otras palabras clave que se mencionan en los artículos que aparece la palabra clave que indica el usuario.

Capítulo 5

Resultados

El sistema que se presenta en esta tesis se desarrolla en varias etapas, las cuales se han descrito a profundidad en capítulos anteriores. Cada etapa del sistema arroja resultados diferentes y la eficacia de cada etapa varia de acuerdo a los diversos problemas y limitaciones que se manifiestan durante su ejecución. A continuación se presentan los resultados de las etapas más importantes del sistema, describiendo el desempeño del procesamiento de información que corresponde a cada etapa, así como los inconvenientes que tuvieron que resolverse o las limitaciones que el sistema tiene en cada etapa.

La primera sección de este capítulo presenta los primeros resultados parciales del sistema de la extracción de datos en artículos científicos en formato PDF utilizando GATE, implementando reglas en Jape y desarrollando algunos métodos en Java, así como las ventajas y desventajas que tiene el sistema al haber basado sus reglas de extracción en cierto tipo de estilos de documentos.

Los datos que se desea extraer de un documento son título, autores, resumen y palabras clave, y para cada uno de ellos se implementó una regla, que a su vez utiliza la etiqueta de otras reglas que identifican otros datos auxiliares (ver capítulo 4.1). Los resultados de las reglas que identifican los datos auxiliares no se reportan de forma individual, ya que estos se ven reflejados en los resultados de las reglas que identifican el título, autores, resumen y palabras clave.

Otra etapa importante del sistema es el desarrollo del método para la comparación de oraciones, mediante la cual se obtiene un grado de similitud entre dos oraciones, lo es utilizado para seleccionar qué documentos tienen oraciones relacionadas (ver capítulo 4.4). En la segunda sección de este capítulo, se presentan los resultados del método implementado para la comparación de oraciones, así como las ventajas y desventajas de apoyarse de este tipo de criterios para generar información sobre grupos de artículos y autores.

En la última sección de este capítulo, se presenta un panorama general del desempeño del

sistema para generar información, y la posibilidad de que el usuario pueda recuperar información sobre las áreas de trabajo de un autor y la relación que tiene con otros autores, así como los documentos que pueden ser de interés sobre un tema determinado.

5.1. Extracción de datos de artículos científicos

La extracción de datos de documentos se llevó a cabo mediante el anotado de documentos con GATE, el cual identifica los datos deseados a través de reglas implementadas con Jape y en algunos casos los datos extraídos pasan por otro proceso para eliminar caracteres o separar anotaciones que es difícil procesar mediante las reglas. Los datos que se identifican son correo electrónico, universidad, sección, título, autores, resumen y palabras clave.

La tabla 5.1 muestra los resultados que se obtuvieron en la extracción de datos de los documentos. En la primera columna se indica el tipo de dato que extrae del documento, mientras que en la segunda columna se presenta el porcentaje de artículos de los cuales se pudo extraer correctamente el dato. Para reportar estos resultados se analizaron artículos con formatos de las revistas *IEEE*, *ACM* y *Springer*. Sin embargo, cabe destacar que a pesar de que cada una de las revistas sugiere cierto tipo de formato para la presentación de los artículos que pública, es común encontrar algunos documentos que no se encuentran en tal formato.

Tipo de dato	**Datos extraídos**
Título	88 %
Autores	86 %
Resumen	93 %
Palabras clave	97 %
Correo eléctronico	84 %
Universidad	77 %

Cuadro 5.1: Porcentajes de datos recuperados de la extracción de datos de documentos.

La regla que se implementó en Jape para identificar el correo electrónico funciona muy bien para el formato más común en que se presenta esta información, que es una palabra o serie de caracteres seguidos del símbolo arroba y otra serie de caracteres, otro formato en el que se puede presentar el correo electrónico es un símbolo de inicio como un < seguido de varias palabras separadas por comas, seguidas de un símbolo de fin como un > seguido del arroba y otra serie de caracteres, en este caso la regla también funciona bien, sin embargo, para identificar cada correo electrónico no es suficiente una regla en Jape, en este caso es necesario utilizar el método implementado en Java que separe las palabras y forme los correos electrónicos verdaderos para ser asignados a cada autor. En la tabla 5.2 se muestran los ejemplos de estos dos formatos.

A pesar de que el correo electrónico se presenta de forma muy estructurada y se podría decir

Dato etiquetado por GATE	Dato final mejorado
palabra@dominio	palabra@dominio
<palabra1,palabra2,palabra3>@dominio.com	palabra1@dominio palabra2@dominio palabra3@domino

Cuadro 5.2: Formatos de correos electrónicos reconocidos.

que es sencillo identificarlo, en algunas ocasiones el correo electrónico no está separado del autor mediante un espacio o la serie de correos no contiene el símbolo de arroba lo que hace imposible identificarlo. Otro problema que se puede presentar con el correo es asignarlo correctamente a su autor, es común que un autor y sus datos se encuentren en la parte superior del documento después del título, sin embargo, cabe señalar que GATE debe identificar el título en una columna, y en seguida identificar los autores que pueden presentarse en varias columnas que contienen el nombre del autor con sus datos, estas columnas puede presentarse alineadas o en algunos casos raros esparcidas o demasiado juntas, en un caso normal GATE transforma estas columnas a párrafos, lo que facilita su lectura y hace posible identificar los datos deseados, pero en otros casos GATE desordena la información y a pesar de que los datos se etiqueten correctamente, el orden correcto de los datos se pierde lo que perjudica la asignación del correo al autor que le corresponde.

La identificación de las columnas en el documento es un problema que repercute en la identificación de otros datos como institución, autor y resumen. En el caso de los datos universidad y autor el problema es el mismo que con el correo electrónico, sin embargo, el problema se presenta menos debido a que sólo sucede cuando las columnas están demasiado juntas y GATE no identifica que se trata de columnas, de modo que cuando dos o más autores están separados sólo por espacios grandes, no es posible identificar que se trata de varios autores en un mismo renglón, ya que GATE transforma el espacio grande en un espacio de un carácter, mientras que el problema con el dato institución tiene el mismo comportamiento que el correo electrónico. La figura 5.1 muestra un ejemplo de como se presentan los autores de un artículo [46] con sus datos en columnas separadas por una delgada linea punteada sólo para hacer evidente como GATE reconoce estas columnas, en el ejemplo los datos de los autores están suficientemente deparados para que GATE reconozca las columnas, sin embargo, de haber menor separación las columnas no se podrían identificar, esto puede presentarse en artículos con más de tres autores. A pesar de los problemas de GATE para transformar el artículo en un texto simple, el correo electrónico se puede obtener en la mayoría de los documentos procesados.

El dato institución es etiquetado por la identificación de palabras reservadas como “University”, “Institute” y otras, por lo que está sujeto a que la palabra que identifica a la universidad como un instituto se encuentre en la lista de palabras que la regla de GATE utiliza. La lista de palabras puede incrementar para mejorar el etiquetado del nombre de un instituto, sin embargo, las palabras pueden ser muy variadas debido a que no hay una norma para llamar a un instituto y además esta palabra puede estar escrita en cualquier idioma. A pesar de los problemas, la universidad o instituto

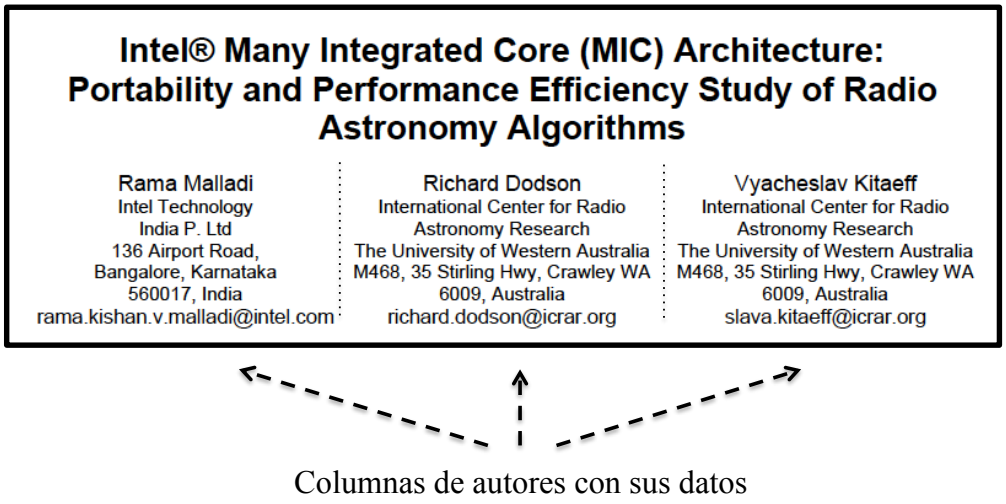

Figura 5.1: Ejemplo de un artículo con autores y datos de autores presentados en columnas.

de un autor puede ser identificado en la mayoría de las ocasiones.

El autor es uno de los datos más importantes que se desea obtener y puede ser identificado principalmente por ser de los primeros datos que aparecen en documento, son nombres propios e inician con mayúscula, entre otras características que se incluyen en la regla implementada en Jape. Otro problema que se presenta para etiquetar los autores de un documento además del que se ha mencionado relacionado con la identificación de columnas por parte de GATE, es el problema de separar el nombre del autor de otros datos. La regla que se implementó para etiquetar el nombre del autor pudo haberse implementado de forma menos flexible para etiquetar asegurar que no se filtraran datos adicionales, sin embargo al realizar las pruebas de la regla, también aumentaba la perdida de nombres de autores, por lo que la regla se hizo menos flexible y en un proceso posterior eliminar los datos que no son el nombre del autor.

De modo que la regla que etiqueta el autor identifica en su gran mayoría el nombre de autores, correos electrónicos y universidades o institutos, este no es un buen resultado, pero es mejor que perder totalmente el nombre del autor, mediante el método implementado en java es que finalmente se obtiene el nombre del autor, obteniendo el resultado reportado en la tabla 5.1. La tabla 5.3 muestra algunos ejemplos de los formatos en que se pueden presentar los nombres de autores y el sistema identifica.

Por otra parte, la recuperación del título es sencilla si se piensa que es el primer renglón que aparece en el documento, sin embargo hay que tener en cuenta que en algunos documentos aparece un encabezado, que en algunas ocasiones GATE excluye del documento, pero en otras no lo hace y la regla excluye este dato. Los problemas que se pueden presentar son cuando los títulos incluyen símbolos, los cuales podría incluirse en la regla para que se acepten como parte del título, pero esto en cambio puede perjudicar la selección de títulos ya que podría parecer cualquier renglón en el

Formato	Ejemplo	Dato final mejorado
Name,[1]	Joan Boyar,[1]	Joan Boyar
Name, Universidad	Joan Boyar, Institute for Integrative Nutrition	Joan Boyar
Name, email	Joan Boyar, jboyar@nutrition.com	Joan Boyar
Name, email,[2]	Joan Boyar, jboyar@nutrition.com,[2]	Joan Boyar
Name	Joan Sebastian Boyar Navarro	Joan Sebastian Boyar Navarro
Name	Joan S. Boyar	Joan S. Boyar
Name	Joan Boyar-Navarro	Joan Boyar-Navarro
Name	Martín Boyar	Martín Boyar
Name	Vincent van Duysen	Vincent van Duysen
Name, Name, Name	Joan Boyar, Vincent van Duysen and Martín Navarro	Joan Boyar Vincent van Duysen Martín Navarro

Cuadro 5.3: Ejemplos de formatos en que se presentan los autores y el sistema recupera.

documento, además el título es uno de los datos que se extrae más exitosamente.

El dato que ha sido llamado sección, no es un dato que se almacene en la base de datos ni en la ontología, sin embargo ayuda a resolver algunos problemas y mejorar el resultado para obtener el resumen y filtrar otros datos que no deben ser reconocidos como autores, datos que son nombres propios pero se encuentran dentro de texto, o bien para limitar donde termina la lista de palabras clave que comúnmente no son terminan con un punto final. El porcentaje de las secciones que se recuperan no se reporta en la tabla 5.1 debido a que sólo es un dato etiquetado de forma auxiliar, sin embargo cabe mencionar que se trata de un dato muy fácil de identificar y se recupera aproximadamente el 95 % de las veces, el problema que evita que se recupere el dato es que GATE no codifique el símbolo extraño que contenga la sección.

El resumen es otro de los datos que se recupera más fácilmente, el etiquetado que hace GATE con la regla correspondiente es sencilla debido a que sólo identifica la palabra “Abstract” y algunas otras variantes, por lo que esta regla quizá etiqueta aproximadamente el 100 % de estas palabras que marcan el inicio del resumen, sin embargo, este no es el dato real que se desea obtener, en realidad lo que se quiere es el texto que contiene el resumen del artículo, por lo que con la ayuda del etiquetado de la sección y un proceso posterior con métodos implementados en Java se obtiene este texto, cuyo resultado es reportado en la tabla 5.1. Los problemas que ocasionan que este texto no se recupere, es que éste no exista en el documento, o que la sección no tenga una marca de inicio o un título de sección.

El inicio de la lista de palabras clave de un artículo está marcado en su gran mayoría por algunas palabras reservadas como “keyword” o “Index terms” y otras variantes, esto hace que las

palabras clave se etiqueten fácilmente mediante una regla que utilice GATE, el problema se presenta para limitar donde terminan, lo cual se resuelve con la ayuda del dato sección, los resultados para identificar las palabras clave son muy buenos, por lo que pueden tomarse para clasificar un documento.

5.2. Comparación de oraciones

La comparación de oraciones es una etapa del procesamiento de documentos muy importante en el sistema, por que en ella se apoya una parte fundamental para poblar la ontología, a pesar de que dos artículos pueden compartir un grupo de palabras iguales en su contenido, no es razón suficiente para ser artículos similares. La similitud de las oraciones que están en los documentos es el requisito final para relacionar dos artículos. De modo que, se propuso un método para comparar oraciones y aprobar la similitud entre oraciones, por eso es importante identificar cuál es el grado de similitud que indica que las oraciones tienen una relación.

El método propuesto intenta identificar oraciones con información similar. El enfoque del método es diferente a otros que tratan de encontrar una oración dentro de otra para identificar o complementar información, porque este método no sólo busca una oración contenida en otra, sino que busca la similitud entre dos oraciones, siendo relevantes las diferencias entre las oraciones. La tabla 5.4 muestra algunos ejemplos de comparación de oraciones, las cuales son comparadas con la primer oración de la tabla, estas comparaciones se hacen a partir de sustantivos que aparecen en las dos oraciones que se comparan. Las primeras oraciones en la tabla 5.4 son muy similares porque ellas cambian con respecto a la oración que se compara sólo en el tiempo gramatical. Por ejemplo, para la oración que está escrita en pasado da una similitud de 1, lo cual es lógico porque el método intercambia toda las palabras por su variedad morfológica, de modo que las palabras “gives” y “gave” son la misma después de aplicar los primeros pasos del método. Según los criterios del método el tiempo gramatical de la oración no afecta su significado. Las oraciones en los siguientes renglones de la tabla tienen tienen información diferente y se puede ver que el grado de similitud disminuye.

En las oraciones de la tabla 5.4 las palabras que se repiten son “experiment” y “result”, las cuales pueden relacionarse fácilmente en una oración, por lo que a pesar de que se agregue o se elimine información, la oración contra la que se comparan las oraciones de la tabla está completamente o en parte contenida en las otras. La tabla 5.5 muestra oraciones con las palabras “Cheetah” and “world”, las cuales pueden tener menos relación en una oración. La primer oración de la tabla 5.5 es la oración que se compara con las otras oraciones que muestra esta tabla. Los grados de similitud que se obtienen comparando estas oraciones son más bajos que los grados de similitud obtenidos en comparaciones de la tabla 5.4, lo cual es una observación importante porque a pesar de que las oraciones contienen también palabras iguales, las palabras no se relacionan lo suficiente para alcanzar un valor de similitud mayor.

Oración	**Grado de similitud**
The experiment gives satisfactory results	1
The experiment is giving satisfactory results	0.9
The experiment gave satisfactory results	1
The experiment was giving satisfactory results	0.9
The experiment has given satisfactory results	0.9
The experiment has been giving satisfactory results	0.84
The experiment had given satisfactory results	0.9
The experiment had been giving satisfactory results	0.84
The experiment is giving satisfactory results	0.9
The experiment will give satisfactory results	0.91
The experiment is going to give satisfactory results	0.78
The experiment can give satisfactory results	0.91
The experiment could give satisfactory results	0.91
The experiment does not give satisfactory results	0.87
The experiment did not give satisfactory results	0.87
The experiment tries to get different results	0.66
Experiment gives results	0.87
The experiment gets results	0.83
The experiment gives always the same results	0.78
The results were gotten in the experiment	0.48
The experiment has been successful, but we need to get more results	0.54
We have gotten good results, even when the experiment has been complicated	0.54
The experiment was a failure, we will begin another project to get better results	0.52
The set of experiments has been reported failure by their wrong results	0.47

Cuadro 5.4: Ejemplos de comparación de la oración ‘The experiment gives satisfactory results”.

Oración	**Grado de similitud**
Cheetah is the fastest feline in the world	1
There are many species in danger of extinction in the world, cheetah is one of them	0.32
There are many species in the world, cheetah is a feline	0.36
In the world there are approximately 9000 cheetahs	0.48
Cheetah is one of the cats in the world	0.78
Cheetah is one of the fastest animals in the world	0.83
Cheetah is the fastest mammal in the world	0.93

Cuadro 5.5: Ejemplos de comparación de la oración 'Cheetah is the fastest feline in the world".

De la comparación de oraciones que se muestra en las tablas 5.4 y 5.5 se puede observar que dos oraciones pueden ser similares cuando alcanzan un grado de similitud a partir de 0.45, quizás para algunos observadores este valor no sea suficiente, pero un grado de 0.5 o 0.6 asegura mejor que existe similitud entre oraciones.

La tabla 5.6 muestra otros ejemplos de comparación de oraciones, en la primera columna se encuentran las oracionesque se comparan con la primera oración de la tabla, en la segunda columna se encuentra el grado de similitud que el método entrega cuando se toman como palabras principales los sustantivos de las oraciones, partiendo de ahí la comparación, tal como los ejemplos anteriores, en la tercer columna se encuentra el grado de similitud de las oraciones tomando como palabras principales los sustantivos y verbos de la oración. A diferencia de la segunda columna, la tercer columna muestra resultados sobre una comparación de árboles sintácticos de las oraciones que inician el recorrido de los árboles desde hojas que pueden ser no sólo sustantivos (como los resultados de la segunda columna) si no también verbos, dando oportunidad a realizar varios recorridos de los árboles o incluir verbos que sin ellos el requisito de que las oraciones deben tener al menos dos palabras en común en ambas oraciones, no se cumpliría.

Los resultados de la segunda y tercer columna de la tabla 5.6 muestran que en algunas ocasiones tomar en cuanta los verbos de la oración no hace diferencia en la comparación donde sólo se toman los sustantivos, sin embargo, en otras oraciones donde el sujeto de la oración cambia, y las oraciones no cumplen los requisitos para ser comparadas, tomar en cuanta el verbo hace mucha diferencia. Este detalle es útil si el sujeto de la oración no es importante o si se sabe que el sujeto puede ser llamado de diferentes formas, cuando lo que se busca es información de una acción sobre un sujeto tomar en cuánta el verbo es muy útil. Cuando se toma el verbo como una de las palabras principales, además de los sustantivos, el método puede estar realizando más comparaciones de las que necesita porque el resultado es el mismo que tomando en cuenta sólo sustantivos como palabras importantes, sin embargo, este no es un gran problema porque unos cuantos recorridos más de los

árboles no afecta el desempeño del método y es mejor tomar en cuanta los verbos para obtener mejores resultados de la comparación de oraciones. Otras palabras que pueden tomarse como palabras importantes para iniciar los recorridos son los adjetivos, pero esto depende de los textos que se analicen.

Oración	**Similitud tomando sólo sustantivos**	**Similitud tomando sustantivos y verbos**
The proposed filter removes the noise on the signal	1	1
The noise is removed through the new filter	0.42	0.43
It is easy to remove the noise with the proposed filter	0.57	0.57
When we use the new filter the signal loses the noise	0.53	0.53
The filter shows good results removing the noise	0.58	0.63
The removal of the noise is possible the proposed filter	0.47	0.64
The common filters do not remove the noise	0.62	0.65
The algorithm removes the noise	0	0.73
The new method is successful removing the noise	0	0.73
It removes the noise better than the another method	0	0.49

Cuadro 5.6: Ejemplos de comparación de la oración 'The proposed filter removes the noise on the signal".

5.3. Funcionamiento del sistema

En esta sección se presenta el resultado final de la implementación del sistema que está hecho para ser utilizado por un usuario, por lo que la explicación del funcionamiento del sistema se realiza a partir de la interfaz gráfica de usuario. Las tareas que se pueden realizar en el sistema son configuración, consulta de información y análisis de artículos.

5.3.1. Opciones del sistema

La figura 5.2 muestra la ventana principal de la interfaz, donde se puede apreciar que el sistema ofrece realizar tres tareas al usuario. El primer recuadro que se aprecia en la ventana está hecho para abrir las opciones de configuración del sistema, estas opciones se muestran el figura 5.3, donde se observa la ventana que permite al usuario seleccionar el directorio que contiene los recursos de GATE, el archivo que contiene la ontología e introducir el usuario y contraseña para acceder a la base de datos.

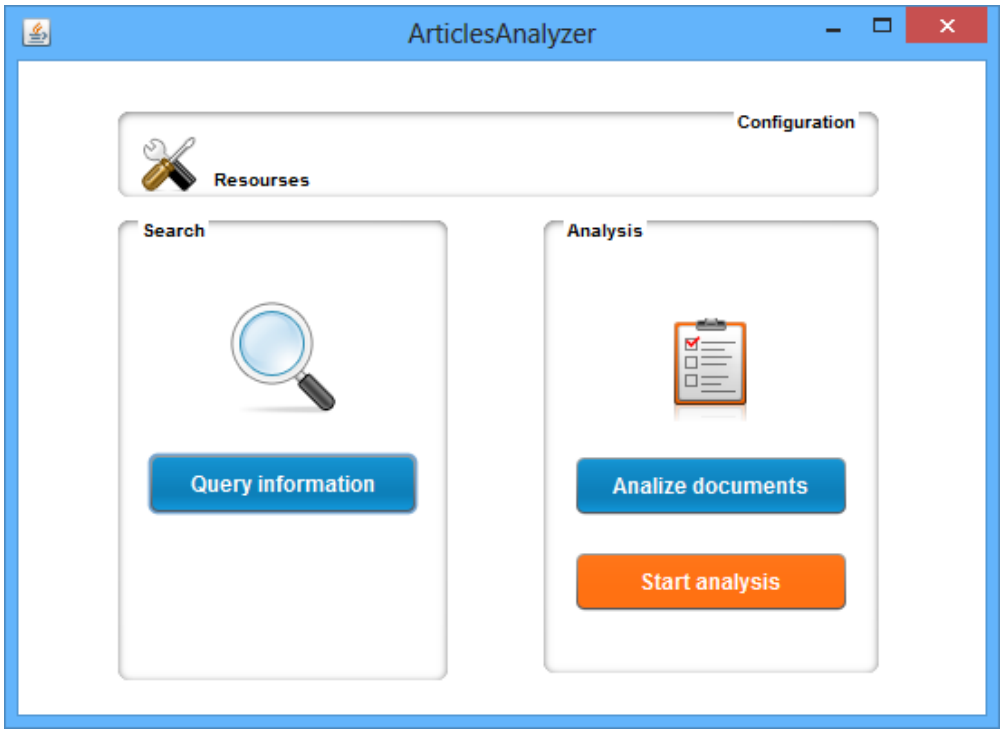

Figura 5.2: Ventana principal de la interfaz gráfica con las tres principales tareas que permite ejecutar el sistema.

El segundo recuadro en el lado izquierdo de la ventana muestra el botón que da inicio a la consulta de información en que contiene la ontología, al presionar este botón el sistema arranca el motor de consultas SQWRL y muestra las opciones de consulta de información. El tercer recuadro en el lado derecho de la ventana muestra dos botones para iniciar el análisis de artículos, el usuario debe presionar el primer botón para seleccionar la carpeta que contiene los archivos que desea que el sistema analice y presionar el segundo botón para iniciar el análisis de los documentos en la carpeta seleccionada.

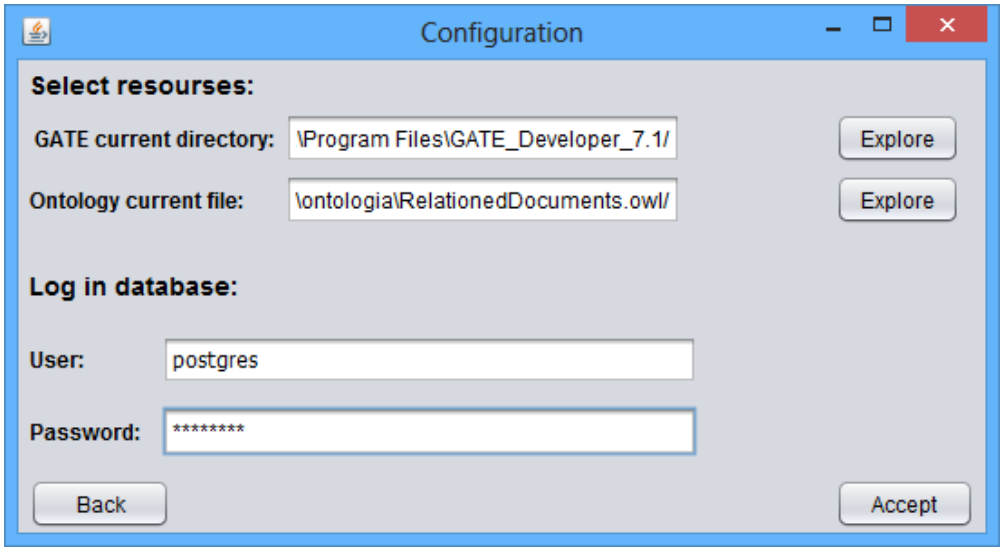

Figura 5.3: Ventana de configuración del sistema.

5.3.2. Relación entre artículos y autores

Las relaciones entre artículos se descubren durante el poblado de la ontología, después de extraer los datos del artículo, hacer varias consultas a la ontología y hacer una comparación de las oraciones en los documentos que probablemente tengan una relación entre ellos y finalmente crear nuevos individuos en la ontología que representan relaciones entre artículos. Mientras tanto, las relaciones entre autores, pueden descubrirse al consultar información de la ontología, buscando qué autores están relacionados por medio de un tema determinado o por trabajar en con otros autores. La información sobre las relaciones entre artículos y autores puede ser consultada a través de la interfaz gráfica.

La figura 5.4 muestra la ventana que aparece al iniciar la búsqueda de información en el sistema. La ventana se compone de tres pestañas, en la primera se muestran las opciones de búsqueda de documentos, en la segunda las opciones de búsqueda de autores y en la tercera las opciones de búsqueda de todos los datos sobre documentos, autores, e instituciones.

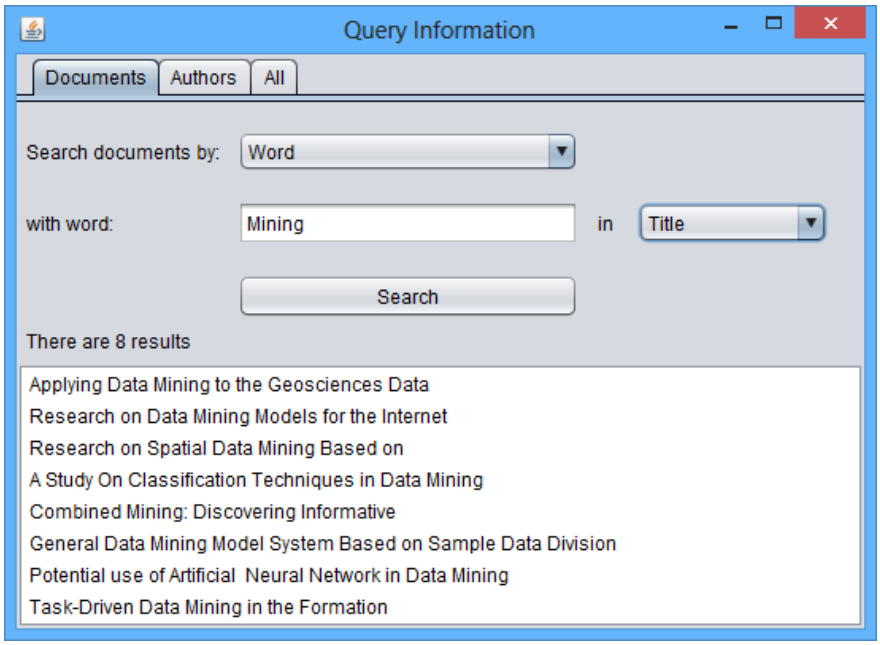

Figura 5.4: Ventana con la pestaña de opciones de búsqueda de documentos.

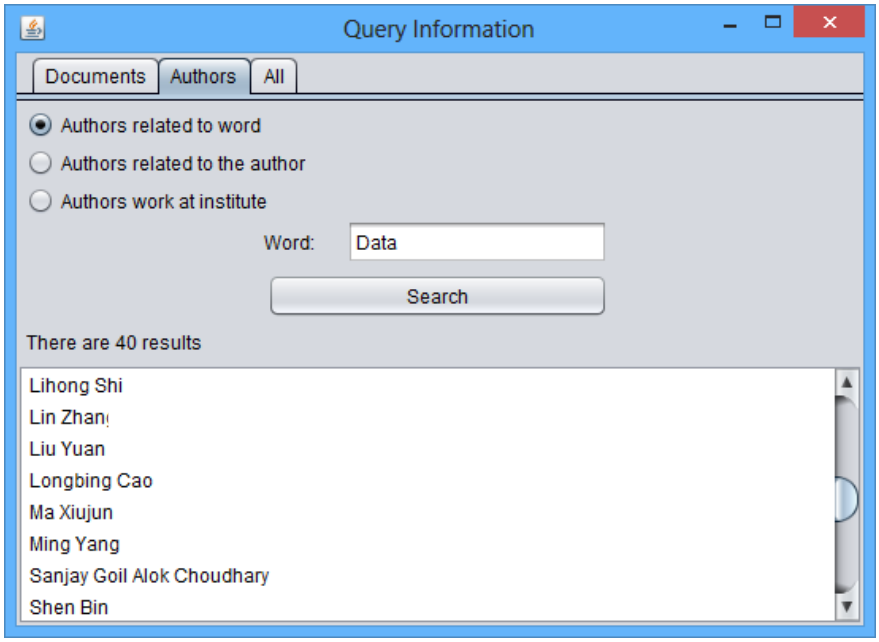

Figura 5.5: Ventana con la pestaña de opciones de búsqueda de autores.

Las opciones de búsqueda de documentos son por palabra, por autor o por instituto, y en caso de ser por palabra, se despliegan otras cuatro opciones que sirven para indicar dónde puede aparecer la palabra indicada por el usuario por ejemplo en el título, palabras clave, resumen o en cualquiera de las tres anteriores.

La pestaña de búsqueda de autores se aprecia en la figura 5.5, esta ventana muestra tres opciones de búsqueda, la primera hace la búsqueda de autores que contengan en sus artículos de investigación palabras indicadas por el usuario, la segunda opción muestra los autores que trabajaron en artículos junto al autor que indique el usuario y la tercera opción muestra autores que trabajan en el instituto que indique el usuario. En la figura 5.6 se aprecia la ventana con las opciones de búsqueda que muestran todos los documentos en el sistema, todos los autores y todos los institutos, así como la opción para buscar una palabra clave relacionada con otra que indique el usuario.

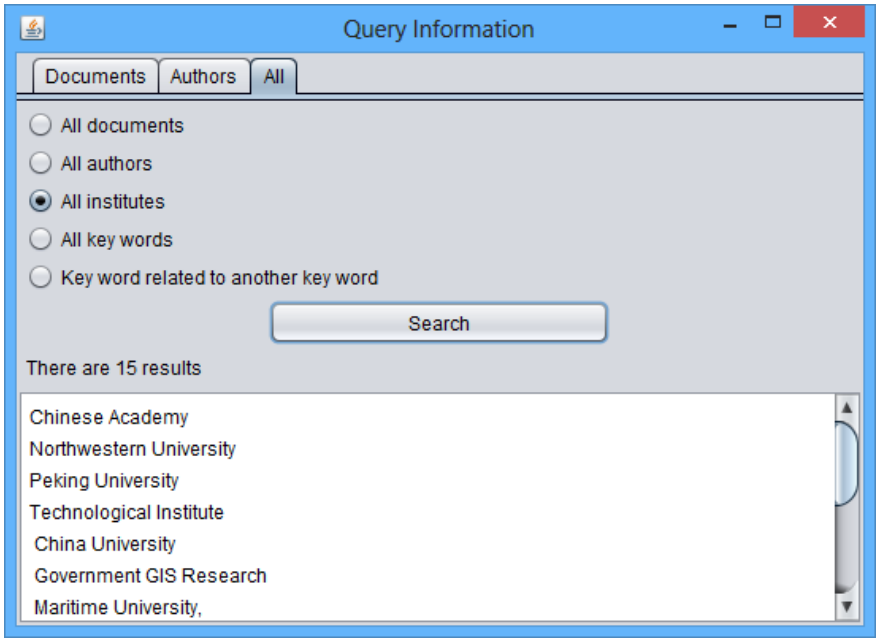

Figura 5.6: Ventana con la pestaña de opciones de búsqueda de todos los datos del sistema.

Las tres pestañas que presentan las diferentes opciones de búsqueda de información en el sistema, entregan cuatro tipos de resultados dependiendo de la consulta, estos resultados pueden ser institutos, autores, documentos y palabras clave. Cualquiera de estos resultados tiene más información sobre ellos mismos, por eso estos resultados son mostrados en vínculos que al hacer doble click sobre ellos abren otra ventana con todos los detalles del resultado.

La figura 5.7 muestra la ventana que despliega los detalles sobre un instituto. En el caso del instituto los detalles que contiene son el propio nombre del instituto y los autores que trabajan en él. Además, la ventana contiene un vínculo para consultar los documentos relacionados al mismo

instituto, al abrir el vínculo se muestra la ventana de consultas de información en la pestaña de búsqueda de documentos para mostrar los resultados, ya que esta pestaña ofrece esta misma opción de búsqueda de documentos relacionados a un instituto.

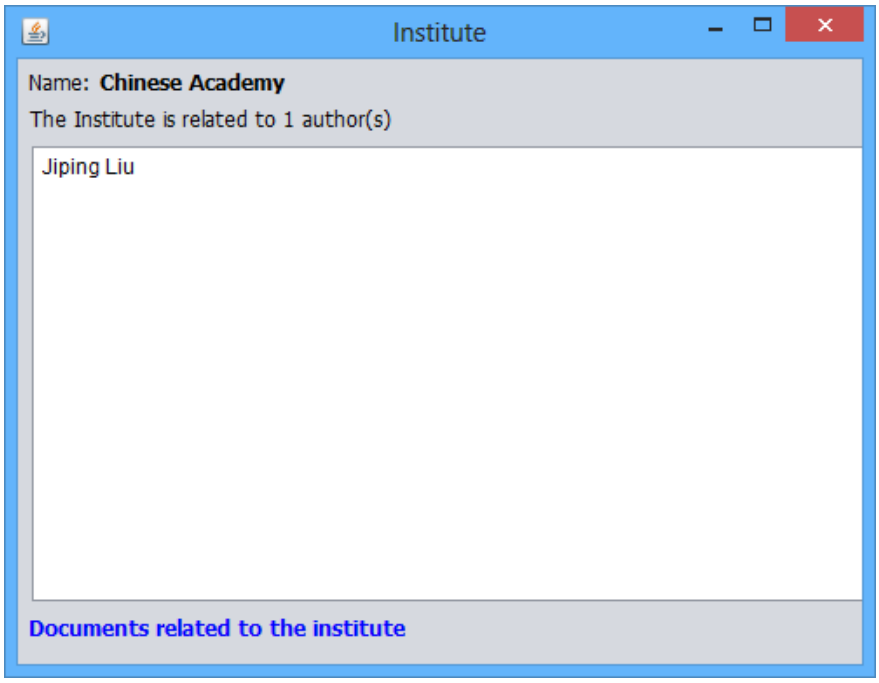

Figura 5.7: Ventana que muestra los detalles de un instituto.

Por otra parte, los detalles de un documento son su título, sus autores, sus palabras clave y su resumen. La figura 5.8 muestra la ventana que despliega los detalles de un documento. Además los autores y palabras clave del documento, se muestran como vínculos, con el fin de poder abrir desde esta ventana los detalles de un autor y palabras clave. Los detalles de un autor, son el propio nombre del autor, su correo electrónico, el instituto donde trabaja y los artículos en los que ha trabajado, de estos datos el instituto es mostrado en un vínculo para mostrar otra ventana con los detalles del instituto. La figura 5.9 muestra la ventana con los detalles de un autor, esta ventana contiene un vínculo para ejecutar una búsqueda de autores relacionados con este autor, cuyos resultados se muestran en la pestaña de búsqueda de autores que también tiene esta opción de búsqueda. Finalmente, la figura 5.10 muestra la ventana con los detalles de una palabra clave, estos detalles son los grupos generales y específicos de palabras relacionados a la palabra clave. Los grupos de palabras son representados como carpetas, un grupo general puede contener grupos específicos y documentos asociados al grupo general, un grupo específico contiene únicamente los documentos.

Figura 5.8: Ventana que muestra los detalles de un documento.

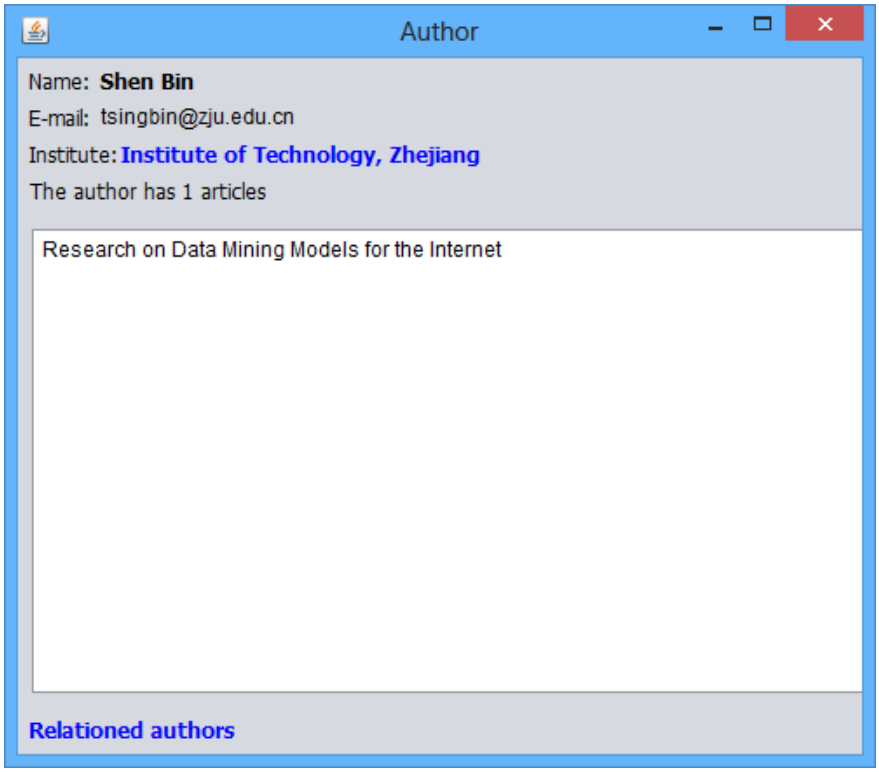

Figura 5.9: Ventana que muestra los detalles de un autor.

Las carpetas son nombradas con una de las palabras que contiene el grupo de palabras que representa. Los documentos además pueden seleccionarse para abrir los detalles del documento. La ventana también contiene un vínculo para ejecutar la búsqueda de otras palabras clave relacionadas a la palabra clave.

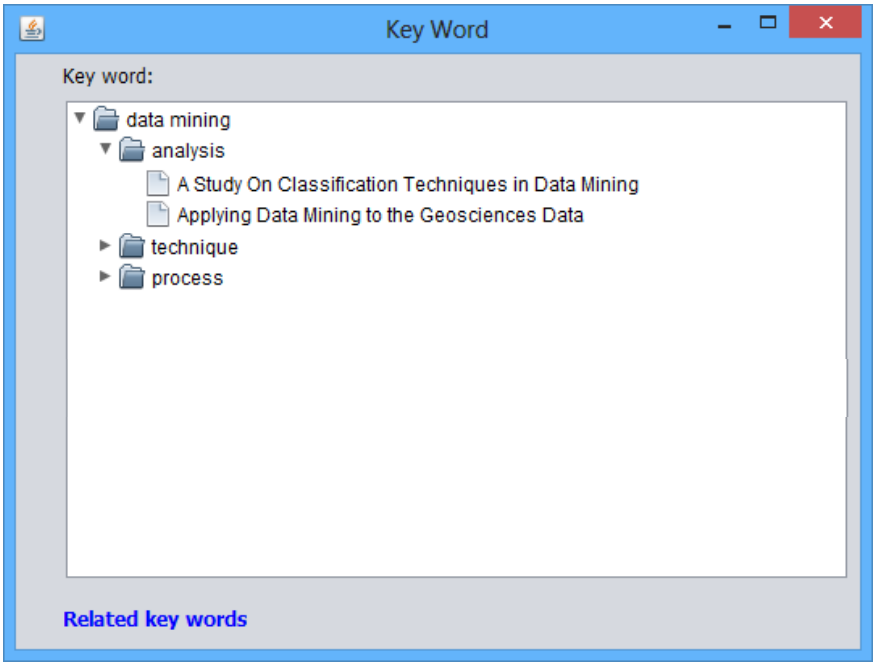

Figura 5.10: Ventana que muestra los grupos de documentos relacionados a una palabra clave.

5.3.3. Análisis de artículos

El análisis de artículos se lleva a cabo en diferentes etapas que no son evidentes para el usuario, debido a que el sistema no muestra el progreso del análisis. Sin embargo, un ejemplo de las etapas de análisis de un grupo de artículos puede apreciarse por medio de la ventana de salida NetBeans, donde se ejecuta el sistema. La figura 5.11 muestra el inicio del proceso donde se inicia GATE, se crea el corpus y se procesan los artículos con los módulos de ANNIE.

La figura 5.12 muestra el resultado de la extracción de datos de uno de los documentos que se procesan, es decir título, resumen, palabras clave y además se puede apreciar la selección de la anotación que marca el final del resumen, donde las siguientes anotaciones ya no son validas. La siguiente etapa en el procesamiento después de la extracción de datos es la creación de individuos en la ontología que representan la relación del artículo que se procesa y otros artículos procesados previamente.

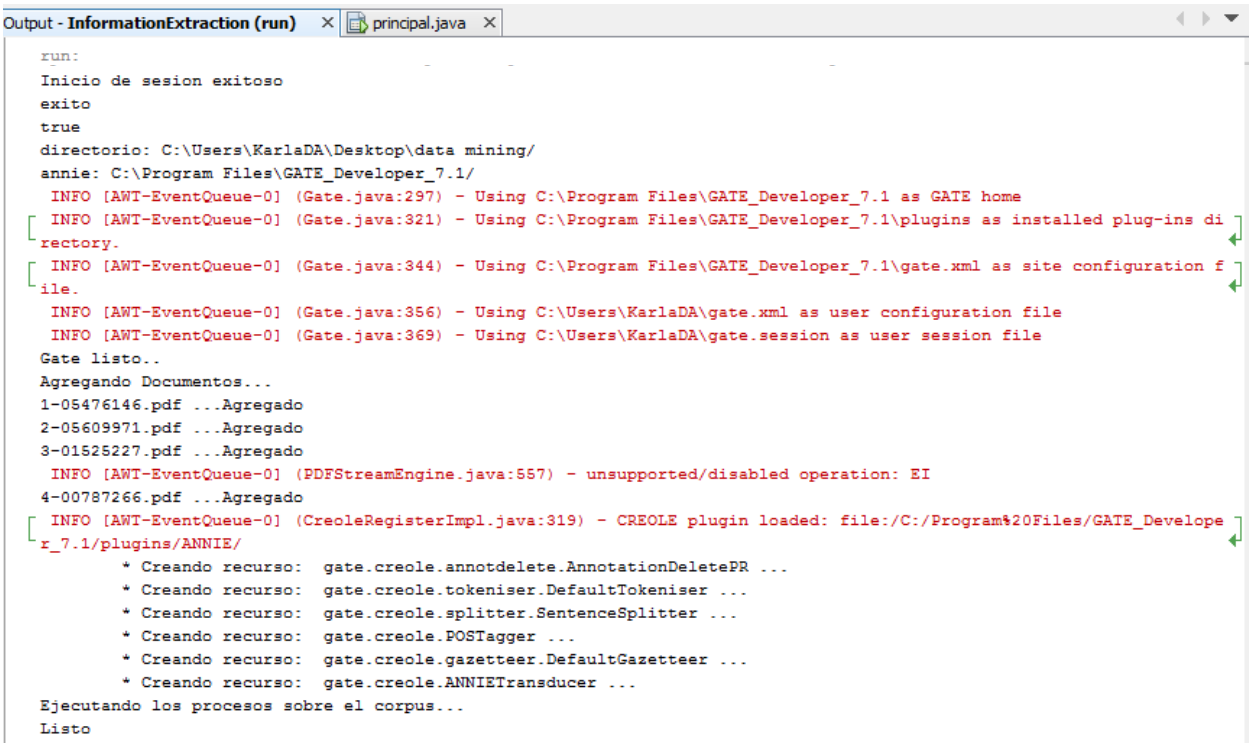

Figura 5.11: Ventana que muestra la ejecución de los módulos de GATE sobre un corpus

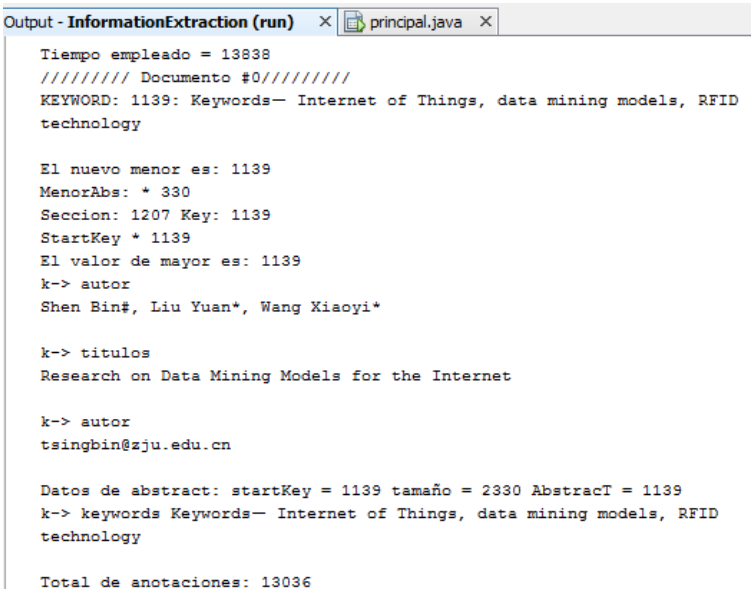

Figura 5.12: Ventana que muestra los resultados del anotado de un documento.

En la figura 5.13 se muestran dos ejemplos del resultado de unas consultas a la ontología, las cuales seleccionan grupos de palabras que probablemente tengan relación con los artículos que se procesan. Cuando se determina que un grupo general de palabras está contenido en el conjunto de palabras del artículo que se procesa, se inicia la comparación de oraciones que contienen las palabras que comparten los documentos asociados al grupo general y el artículo que se procesa. En la figura 5.14 se aprecia el resultado de este proceso y se puede ver que la similitud de sus oraciones es mayor a 50 (en un rango de 0 a 100), lo que significa que los documentos comparados son similares, entonces se crean nuevos individuos de clasificación para representar la relación entre los documentos.

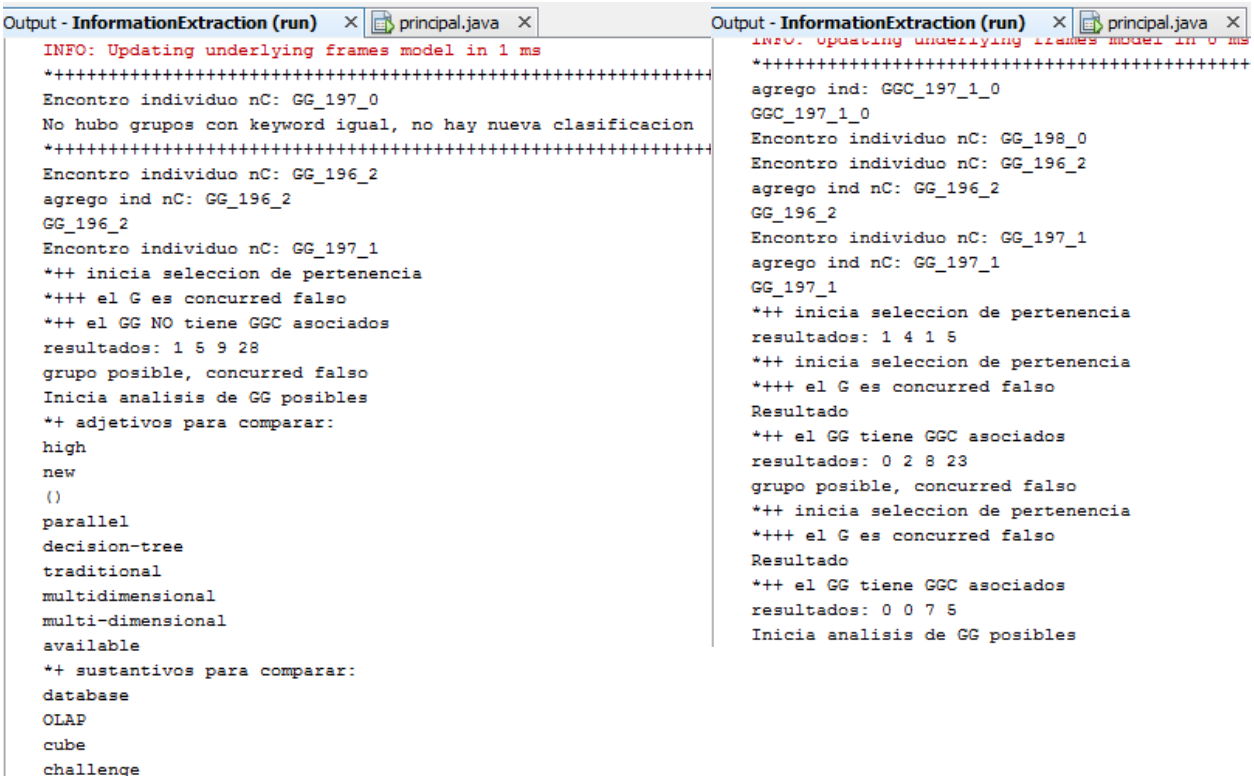

Figura 5.13: Ventana que muestra el resultado de consultas a la ontología para obtener posibles grupos de palabras relacionados a un documento.

Entre todos los documentos que se utilizaron para probar el procesamiento de artículos científicos, sólo un grupo de ellos tiene posibilidad de formar relaciones. El primer requisito para que dos artículos sean relacionados en la ontología, es que compartan una palabra clave, después el grupo de palabras de estos documentos pueden continuar con el proceso de selección hasta determinar con la comparación de sus oraciones si realmente se relacionan. Un ejemplo de grupo de documentos que comparten una misma palabra clave, es un grupo de nueve documentos publicado en IEEE que comparten la palabra clave "data mining". Este grupo de documentos fue procesado por el

sistema y de ellos se obtuvieron tres grupos generales que relacionan hasta tres documentos en un mismo grupo. Esto significa que en primer lugar se formaron tres grupos generales nuevos y posteriormente un documento más se relaciono con el grupo especifico que ya se había formado de la relación de los dos primeros documentos que formaron uno de los grupos. Cabe mencionar que los documentos que se relacionaron en un grupo de palabras, se relacionaron debido a las oraciones en sus resúmenes que mencionan técnicas de minería de datos. A partir de otros grupos de documentos que comparten palabras clave se formaron otros grupos generales con sus propias características, permitiendo comprobar que es posible crear de grupos de documentos que se relacionan entre sí, por medio del análisis de artículos que se implementó.

```
*+ documentos asociados:
doc_196
*++ Inicia comparacion de oraciones
*+ documento con id: 196
*+ Inicia comparacion de frases
Loading parser from serialized file edu/
Loading parser from serialized file edu/
---** Similitud de oracion: 55
Relacion de frases aprobada
Relacion de Grupo aprobada
*++ Inicia crear Grupo General Nuevo
cube
datum
mining
technique
-------- Crea clasificacion -----
-------- Crea clasificacion -----
*++++++++++++++++++++++++++++++++++++++++
```

Figura 5.14: Ventana que muestra los resultados de la comparación de oraciones de dos documentos.

Capítulo 6

Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones y trabajo futuro del desarrollo de esta tesis. En la primera sección se presentan las conclusiones de cada etapa del desarrollo de la tesis, las ventajas y desventajas de haber implementado los métodos propuestos como se hizo, y la conclusión general de la tesis. En la segunda sección se presenta el trabajo futuro que puede realizarse a partir del desarrollo de esta tesis.

6.1. Conclusiones

La extracción de datos de artículos científicos por medio de GATE se lleva a cabo con buenos resultados, la mayoría de los datos son extraídos exitosamente cuando un artículo contiene los datos que se desean, sin embargo, no es posible extraer datos en un ciento por ciento, debido a la gran variedad de formatos en que se escribe un artículo. Es posible identificar una serie de formatos en que los artículos pueden estar escritos, pero siempre es posible que los autores de los artículos no cumplan las recomendaciones de la organizaciones que los publican para escribir sus documentos.

Trabajar con GATE para hacer extracción de datos es sencillo, la interfaz que provee es muy útil para entender y reconocer las anotaciones que se hacen en el documento. A pesar de que GATE cuenta con una gran variedad de módulos para ejecutar diferentes procesos sobre un documento, implementar nuevas reglas que hagan el anotado de nuevos patrones es sencillo. La implementación de nuevas reglas se hace mediante el Jape, que es un lenguaje sencillo pero potente y del cual hay suficiente documentación para utilizarlo, después de crear estas reglas es muy sencillo integrarlas a los módulos de ANNIE. GATE embebido es muy útil porque permite utilizar únicamente los módulos de GATE que uno necesita y extraer fácilmente las anotaciones de los documentos que se procesan, además de que es posible integrar otras funciones por medio de java.

El anotado de un documento se realiza de forma serial conforme se ejecutan las reglas, éstas pueden utilizar las anotaciones que otras reglas ya hicieron en el documento para encontrar nuevas anotaciones, lo que resulta ser una ventaja para implementar nuevas reglas y encontrar anotaciones. A pesar de esto, en ocasiones un conjunto de datos que se quisiera fuera identificado como un conjunto de anotaciones, debe ser etiquetado en una sola anotación, como es el caso de las palabras clave, que son una lista de palabras escritas de forma muy común y durante el anotado del documento, la lista de palabras clave se identifica en una sola anotación y no en varias anotaciones que identifiquen cada palabra clave. Este problema se debe a que en GATE no es posible hacer una nueva anotación sobre otra anotación, lo que es diferente a que dos o más anotaciones etiqueten la misma palabra. Sería deseable que una anotación pudiera ser procesada y que a partir de ella se obtengan nuevas anotaciones, para que por ejemplo se pudiera etiquetar cada palabra clave a partir de la anotación de la lista de palabras clave, en lugar de tener que ser procesada para separar cada palabra clave mediante otros métodos.

Otro problema es el tiempo para hacer el anotado de los documentos porque entre más reglas se utilicen mayor es el tiempo de ejecución de cada documento y un documento puede requerir de varios segundos para ser anotado. Sin embargo, el resultado es muy confiable y en el caso de los artículos científicos es posible obtener datos muy importantes que caracterizan el documento.

Por otra parte, los datos que se extraen de los documentos se guardan en una base de datos y en una ontología. Los datos que se almacenan en la base de datos son las oraciones del documento y hacerlo así dio buenos resultados, debido a que se evitó que la ontología se poblara con datos que no sirven para hacer relaciones entre artículos y autores, además por cada documento se generan muchas oraciones sólo guardando aquellas que están en el resumen, extraer las oraciones de otras secciones del documento o del documento completo genera una gran cantidad de oraciones. Esto es importante porque identificar nuevas relaciones entre artículos puede requerir de mucha memoria. Los datos que se almacenan en la ontología sirven para identificar nuevos grupos de palabras que comparten los documentos. La forma en que se hace una consulta a la ontología para identificar nuevas relaciones se hace por medio de operaciones entre conjuntos, lo que facilita desarrollar la consulta, pero realizar estas operaciones en ocasiones es muy costoso, esto ocasiona que una consulta tenga que ser optimizada, realizando varias consultas para obtener la información completa.

La relación de artículos no se obtiene únicamente por medio de las consultas a la ontología, la similitud de oraciones es el criterio final para relacionar dos documentos. El método propuesto da buenos resultados haciendo comparaciones entre oraciones que se utilizan para probar el método, e incluso aplicándolo a oraciones obtenidas de artículos científicos, esto puede observarse durante el análisis de un documento, donde se comparan varias oraciones de los documentos y se realizan varias comparaciones que sirven para reafirmar o descartar una probable relación de dos documentos que comparten un grupo de palabras. Esta aplicación del método es muy útil porque permite relacionar documentos no sólo por las palabras que comparten, sino por la similitud de sus oraciones, a diferencia de otros enfoques que utilizan relacionan documentos cuando éstos comparten una bolsa de palabras sin saber si estas palabras realmente son utilizadas para expresar ideas similares.

El proceso propuesto en esta tesis muestra buenos resultados para relacionar artículos científicos y sus autores, aportando un nuevo enfoque para relacionar documentos por medio de consultas a una ontología que modela las características y relaciones de artículos científicos y la comparación de oraciones de documentos. El desarrollo de este enfoque también, ha dado como resultado la implementación de métodos para extraer datos de artículos científicos, por medio de reglas en jape y métodos en java, así como el método para comparar oraciones, que también puede ser utilizado en extracción de datos u otras aplicaciones. El problema principal que se presenta durante el análisis de un documento es el tiempo que se utiliza para anotar un documento y crear los arboles sintácticos de cada oración. Sin embargo, este problema pudiera solucionarse optimizando algunos aspectos de la implementación.

6.2. Trabajo futuro

La implementación del sistema puede ser mejorada en algunos aspectos para disminuir el tiempo de procesamiento de un documento y utilizar menos recursos. Estos aspectos pueden mejorarse evitando procesar todo el documento, lo que puede hacerse seleccionando sólo algunas secciones del documento para obtener los datos. En un principio se pensó en sólo procesar la primera o segunda página del artículo donde se encuentras los principales datos que se extraen de los artículos, sin embargo no se realizó porque otras oraciones de otras secciones del artículo pueden ser muy importantes. La selección de estas secciones puede aportar una mejora para obtener un mejor análisis de los documentos. El análisis que actualmente se hace de un documento ha permitido comprobar que es posible hacer buenas relaciones poblando la ontología que se propuso, estas relaciones están formadas por el grupo de palabras que contienen dos o más artículos que además son utilizadas en oraciones similares, sin embargo estos grupos de palabras están ligados a una palabra clave.

Sería muy bueno poder proponer otras palabras que tomen el papel de la palabra clave, de modo que no se tenga que depender de que en un artículo estén indicadas las palabras clave, además aun cuando en el artículo se indiquen estas palabras es posible que el autor del artículo no haya puesto una palabra en la lista de palabras clave, cuando si debería de estar en ella. Las palabras que se pueden proponer podrían seleccionarse de las palabras que aparezcan con más frecuencia en un documento, o quizás buscando palabras que son palabras clave de otros documentos. El objetivo es formar nuevos grupos de palabras que puedan representar nuevos temas de un documento que quizás no son los más importantes pero si tienen relación con otros artículos y pueden aportar nuevas ideas sobre un tema.

La búsqueda de nuevos grupos de palabras requiere de recursos de memoria y tiempo para procesar el documento, que pueden mejorarse como se ha mencionado además, esto no pudo realizarse hasta no asegurarse de que se forman relaciones por medio de grupos de palabras. Ubicar otros grupos de palabras permitiría ofrecerle mayor información al usuario del sistema o quizás obtener información sobre qué temas están relacionados con otros para tener una visión más general sobre

como un problema puede ser resuelto.

Un trabajo que puede realizarse a partir del procesamiento de anotaciones en la extracción de datos es la implementación de un módulo que pueda ser utilizado en GATE para hacer anotaciones exclusivamente sobre otra anotación, o crear un API para GATE embebido que permita hacerlo. Esto sería muy útil para seleccionar datos que son más difíciles de encontrar con reglas que identifican patrones como se hace con jape. Dicho módulo es viable porque GATE es un software libre al que se le pueden agregar nuevos módulos para mejorar su funcionalidad y expandir el procesamiento de lenguajes naturales.

Por otra parte la ontología que se propuso es muy sencilla, debido a que simplemente trata de modelar las relaciones de artículos por medio de conjuntos de palabras que comparten entre sí, sin embargo, esta ontología puede extenderse para modelar mayor información sobre los autores de los artículos u otras personas involucradas en el desarrollo de proyectos de investigación como profesores o alumnos, además se podría modelar otro tipo de documentos como reportes de proyectos, o modelar más características de un instituto. Hacer el modelado de la ontología es muy viable, el problema principal es obtener los datos para poblar la ontología, lo que desafortunadamente no puede hacerse mediante extracción de datos de artículos científicos. Otras ontologías modelan esta información, sin embargo, mediante el análisis de documentos que se presenta en esta tesis, se podría obtener mejores relaciones semánticas entre los individuos de la ontología, que permitan obtener más información que además sea más confiable debido al análisis más exhaustivo oraciones cada documento.

Bibliografía

[1] C. Alejandro Serralde Romero. Recuperación de la información. Technical report, Universidad de Guadalajara, Sistema de Universidad Virtual, Mexico, Guadalajara, 2012.

[2] Ramón Núñez Centella. Galileo, pionero de la divulgación científica. http://sociedad.elpais.com, March 2010.

[3] Juan Carlos Argüelles. ¿qué es la producción científica? http://sociedad.elpais.com, February 2008.

[4] Harith Alani, Srinandan Dasmahapatra, Kieron O'Hara, and Nigel Shadbolt. Identifying communities of practice through ontology network analysis. *IEEE Intelligent Systems*, 18(2):18–25, 2003.

[5] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 990–998, Las Vegas, Nevada, USA, August 2008. ACM.

[6] F. Shaikh, U.A. Siddiqui, I. Shahzadi, S.I. Jami, and Z.A. Shaikh. Swise: Semantic web based intelligent search engine. In *Information and Emerging Technologies (ICIET), 2010 International Conference on*, pages 1–5, Karachi, Pakistan, June 2010. IEEE.

[7] Melania Degeratu and Vasileios Hatzivassiloglou. Building automatically a business registration ontology. In *Proceedings of the 2002 annual national conference on Digital government research*, pages 1–7, Los Angeles, California, May 2002. Digital Government Society of North America.

[8] Valentina Muñoz Porras. Herramientas para la extracción de información para el español acoplables a Gate. Master's thesis, UNAM, Instituto de Investigación en Matemáticas Aplicadas y Sistemas, Septiembre 2008.

[9] S. Zaidi, M.T. Laskri, and A. Abdelali. Arabic collocations extraction using gate. In *Machine and Web Intelligence (ICMWI), 2010 International Conference on*, pages 473–475, Algiers, Algeria, October 2010. IEEE.

[10] M.F. Reza and R. Matin. Application of data mining for identifying topics at the document level. In *Informatics, Electronics Vision (ICIEV), 2013 International Conference on*, pages 1–6, Dhaka, Bangladesh, May 2013. IEEE.

[11] N.M. Sharef and S.A.M. Noah. Semantic search processing in natural language interface. In *Computing and Convergence Technology (ICCCT), 2012 7th International Conference on*, pages 1436–1442, Seoul, Rep. of Korea, December 2012. IEEE.

[12] Michael Uschold and Michael Gruninger. Ontologies and semantics for seamless connectivity. *SIGMOD Rec.*, 33(4):58–64, December 2004.

[13] Yasar Guneri Sahin, Sabah Balta, and Tuncay Ercan. The use of internet resources by university students during their course projects elicitation: A case study. *TOJET: The Turkish Online Journal of Educational Technology*, 9(2):234–244, April 2010.

[14] H. Alani, Sanghee Kim, D.E. Millard, M.J. Weal, W. Hall, P.H. Lewis, and N.R. Shadbolt. Automatic ontology-based knowledge extraction from web documents. *Intelligent Systems, IEEE*, 18(1):14–21, January 2003.

[15] IEEE. Author digital tool box. http://www.ieee.org/publications_standards/publications/authors/authors_journals.html, October 2013.

[16] ACM. Home acm. http://www.acm.org, October 2013.

[17] Springer. Autores and editores. http://www.springer.com/authors?SGWID=0-111-0-0-0, October 2013.

[18] Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. Parsing with compositional vector grammars. In *ACL (1)*, pages 455–465, 2013.

[19] L. Ahmedi, L. Abazi-Bexheti, and A. Kadriu. A uniform semantic web framework for co-authorship networks. In *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*, pages 958–965, Sydney, Australia, January 2011. IEEE.

[20] Dan Brickley. The friend of a friend (foaf) project. http://www.foaf-project.org, February 2014.

[21] York Sure, Stephan Bloehdorn, Peter Haase, Jens Hartmann, and Daniel Oberle. The swrc ontology - semantic web for research communities. In Carlos Bento, AmÃlcar Cardoso, and Gaël Dias, editors, *Proceedings of the 12th Portuguese Conference on Artificial Intelligence - Progress in Artificial Intelligence*, pages 218–231, Covilhã, Portugal, December 2005. Springer.

[22] Informatics institute of the faculty of economics and business engineering. http://www.aifb.kit.edu/web/SWRC/en, February 2014.

[23] Ontoware shutdown. http://www.ontoware.org, February 2014.

[24] Ontowabe. http://ontoweb-lt.dfki.de, February 2014.

[25] Arnetminer. Aminer. http://arnetminer.org, October 2013.

[26] Ismael Alberto Cruz Vargas. Sistema de procesamiento y clasificación de textos de investigación. Technical report, Universidad Autónoma Metropolitana, Unidad Azcapotzalco, Distrito Federal, Mexico, Julio 2013.

[27] Daniel Micol, Óscar Ferrández, Rafael Muñoz, and Manuel Palomar. Dlsite-2: Semantic similarity based on syntactic dependency trees applied to textual entailment. In *United States of America*, pages 73–80, Rochester, NY, USA, April 2007. Association for Computational Linguistics.

[28] Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, pages 423–429, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[29] Rui Wang and Günter Neumann. Recognizing textual entailment using sentence similarity based on dependency tree skeletons. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, pages 36–41, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[30] Katrin Fundel, Robert Küffner, and Ralf Zimmer. Relex - relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, January 2007.

[31] Asociación Mexicana para el Procesamiento del Lenguaje Natural. ¿qué es procesamiento del lenguaje natural? http://www.ampln.org/pmwiki.php?n=Main.PLN, Octubre 2013.

[32] Alejandro Peña Ayala. *Lenguaje Natural: Descripción de las Etapas para su Tratamiento.* APA, México, D.F., first edition, 2006.

[33] Mari Vallez and Rafael Pedraza Jimenez. El procesamiento del lenguaje natural en la recuperación de información textual y áreas afines. "hipertext.net", núm. 5, 2007. http://www.upf.edu/hipertextnet/numero-5/pln.html, Octubre 2013.

[34] Eduardo Sosa. Procesamiento del lenguaje natural: revisión del estado actual, bases teóricas y aplicaciones. *Information world en español*, 6(1-2):26–29, January 1997.

[35] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press., Cambridge, MA USA, first edition, 1999.

[36] Mark Sanderson. Retrieving with good sense. *Information Retrieval*, 2(1):49–69, 2000.

[37] E. Marsh and D. Perzanowski. Evaluation of ie technology: Overview of results. In *Message Understanding Conference Proceeding, MUC-7*, pages 5–8, San Diego, CA USA, April 1998.

[38] Azucena Jiménez Pozo. Adaptación y mejora de un sistema de preprocesamiento y categorización gramatical. Master's thesis, Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad Politécnica de Madrid, December 1999.

[39] Zhang Zhixiong, Li Sa, Wu Zhengxin, and Lin Ying. Towards constructing a chinese information extraction system to support innovations in library services. In *97 Information Technology with Audiovisual and Multimedia and National Libraries*, pages 1–16, Beijing , China, 2006. The library of Chinese Academy of Sciences.

[40] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and et al. Developing language processing components with gate version 7 (a user guide). Technical report, The University of Sheffield, Sheffield, England, November 2013.

[41] Beatrice Santorini. Part-of-speech tagging guidelines for the penn treebank project. http://www.cis.upenn.edu/ treebank/, May 2014.

[42] Michael Grüninger and Jintae Lee. Ontology applications and design - introduction. *Commun. ACM*, 45(2):39–41, 2002.

[43] Yip Chi Kiong, S. Palaniappan, and N.A. Yahaya. Health ontology system. In *Information Technology in Asia (CITA 11), 2011 7th International Conference on*, pages 1–4, Sarawak, Malaysia, July 2011. IEEE.

[44] B Chandrasekaran, J.R. Josephson, and V.R. Benjamins. What are ontologies, and why do we need them? *IEEE Intelligent Systems*, 14(1):20–26, 1999.

[45] Yao-Tang Yu and Chien-Chang Hsu. A structured ontology construction by using data clustering and pattern tree mining. In *International Conference on Machine Learning and Cybernetics, ICMLC 2011*, pages 45–50, Guilin, China, 2011. IEEE.

[46] Rama Malladi, Richard Dodson, and Vyacheslav Kitaeff. Intel many integrated core (mic) architecture: Portability and performance efficiency study of radio astronomy algorithms. In *Proceedings of the 2012 Workshop on High-Performance Computing for Astronomy Date*, AstroHPC '12, pages 5–6, New York, NY, USA, 2012. ACM.