UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

RESÚMENES SEMIAUTOMÁTICOS DE CONOCIMIENTO: CASO DE RDF

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

CAMILO FERNANDO GARRIDO GARCÍA

PROFESOR GUÍA:
CLAUDIO GUTIÉRREZ GALLARDO

MIEMBROS DE LA COMISIÓN:
BÁRBARA POBLETE LABRA
JORGE PÉREZ ROJAS

SANTIAGO DE CHILE
ENERO 2013

Resumen

En la actualidad, la cantidad de información que se genera en el mundo es inmensa. En el campo científico tenemos, por ejemplo, datos astronómicos con imágenes de las estrellas, los datos de pronósticos meteorológicos, los datos de infomación biológica y genética, etc. No sólo en el mundo científico se produce este fenómeno, por ejemplo, un usuario navegando por Internet produce grandes cantidades de información: Comentarios en foros, participación en redes sociales o simplemente la comunicación a través de la web.

Manejar y analizar esta cantidad de información trae grandes problemas y costos. Por ello, antes de realizar un análisis, es conveniente determinar si el conjunto de datos que se posee es adecuado para lo que se desea o si trata sobre los temas que son de nuestro interés. Estas preguntas podrían responderse si se contara con un resumen del conjunto de datos. De aquí surge el problema que esta memoria abarca: Crear resúmenes semi-automáticos de conocimiento formalizado.

En esta memoria se diseñó e implementó un método para la obtención de resúmenes semiautomáticos de conjuntos RDF. Dado un grafo RDF se puede obtener un conjunto de nodos, cuyo tamaño es determinado por el usuario, el cual representa y da a entender cuáles son los temas más importantes dentro del conjunto completo. Este método fue diseñado en base a los conjuntos de datos provistos por DBpedia. La selección de recursos dentro del conjunto de datos se hizo utilizando dos métricas usadas ampliamente en otros escenarios: Centralidad de intermediación y grados. Con ellas se detectaron los recursos más importantes en forma global y local.

Las pruebas realizadas, las cuales contaron con evaluación de usuarios y evaluación automática, indicaron que el trabajo realizado cumple con el objetivo de realizar resúmenes que den a entender y representen al conjunto de datos. Las pruebas también mostraron que los resúmenes logran un buen balance de los temas generales, temas populares y la distribución respecto al conjunto de datos completo.

"Como un oso en primavera."

-Haruki Murakami. "Norwegian Wood".

Agradecimientos

Primero quiero agradecerle a mi familia, no por la memoria en sí, sino por por las pequeñas cosas. Por eso le agradezco a mi papá por preocuparse al inicio de cada semestre de que yo tuviera lápiz y goma; a mi mamá por tenerme almuerzo y acompañarme en el desayuno todos los días; y a mi hermano por revisar la ortografía de este informe.

Dentro del contexto de la memoria agradezco a mi profesor guía por su ayuda en el desarrollo de este trabajo; a mi profesora co-guía por la ayuda en la determinación del tema de esta memoria; y al profesor integrante por revisar y encontrar todos los errores en el informe.

Le doy gracias a mis amigos: Al que me dio una canción hace mucho tiempo y que utilizo para animarme; al que escuchó un problema; a la que rayó mis cuadernos y libros; al que me copió y ayudó a sacar proyectos adelante; al que convertía algo cotidiano en algo interesante; y al que me dijo una vez 'si todo fuera tan fácil, no sería interesante'.

Mención especial a la música por ayudarme a ignorar a la gente y así poder trabajar tranquilo. También a las segundas onces por alimentarme y mantenerme despierto. Y a los miércoles de Gorbea, los jueves de café, los lunes de auxiliar, los martes de memoria y viernes de taller.

Tabla de contenido

Capítulo 1

Introducción

Cada vez hay más información, cada vez se están generando más datos. Desde el mundo científico se generan datos biológicos, astronómicos, meteorológicos, geofísicos, etc. Desde el mundo empresarial datos de las bolsas de comercio, transacciones de bancos. Desde la sociedad datos de previsiones, salud, impuestos. Desde la web datos como páginas, documentos, logs de accesos, redes sociales, etc. Esto es lo que se ha llamado *diluvio de datos*. La generación y aparición de grandes cantidades de información gracias a que nuestra capacidad de producirla va aumentando cada vez más.

La cantidad de datos producida por el *diluvio de datos* ha traído problemas, tanto en el almacenamiento como en el procesamiento de ellos, que se han ido superando poco a poco gracias al avance de la tecnología. No obstante, aún quedan problemas pendientes y/o problemas que necesitan muchos recursos para ser resueltos. Por ejemplo, realizar respaldos de aquellos datos requiere muchos recursos, o si se desea realizar un análisis sobre ellos tomaría mucho tiempo o se necesitaría disponer de una elevada capacidad de procesamiento. Esto produce que los datos no puedan ser fácilmente manejados por cualquiera que lo desee. Si no se puede hacer un respaldo de todos los datos entonces se desearía hacer un respaldo de cierta parte, si no se puede realizar un análisis detallado sobre todos los datos se desearía al menos realizarlo sobre una cierta cantidad. ¿Pero sobre qué datos? ¿Cómo seleccionamos que sección de los datos guardar o sobre cuál realizar el análisis? Se desearía escoger aquellos datos que representen mejor o que nos den la mayor cantidad de información o aquellos que sean más relevantes en el conjunto, es decir, realizar un resumen de los datos.

Todo el proceso de aumento de generación de datos y los problemas que conlleva suceden en el contexto de la web. La web tiene como objetivo ser un espacio para compartir información. Está conformada por una colección de documentos de texto, entre otros recursos, enlazados entre sí. Esto forma una gran red de información intercambiable. Por ello se ve directamente afectada con el crecimiento en la generación de datos. Este crecimiento hace que esta red crezca y sea más difícil navegar a través de ella.

La Web Semántica es un movimiento colaborativo liderado por el *World Wide Web Consortium*[3] (abreviado W3C) que promueve el uso de un formato común para la información en la web con el objetivo de que los datos puedan ser compartidos, reusados por aplicaciones

y leídos y navegados por una máquina. Se basa en la idea de añadir metadatos semánticos a la información publicada. Esta está construida sobre la especificación RDF.

El marco de descripción de recursos (RDF por sus siglas en inglés)[2] es un modelo estándar propuesto por la W3C para el intercambio de datos en la web que representa la información de los recursos en la web. RDF amplía la estructura de enlace de la web para usar URI para nombrar la relación entre dos recursos. Usando este modelo simple, permite a datos estructurados y semi-estructurados ser mezclados, expuestos y compartidos a través de la web y distintas aplicaciones. Esta estructura de enlaces forma un grafo dirigido y etiquetado, donde un arco representa el enlace entre dos recursos que son representados por los nodos del grafo. Esta visión de grafo es la forma más fácil de poder pensar el modelo y por lo mismo es utilizado para realizar visualizaciones.

1.1. Problema

En esta memoria se pretende abordar el problema de conocer cuales son los temas o asuntos que describen mejor un conjunto de datos.

Digamos, por ejemplo, que tenemos un libro. Antes de leerlo querríamos saber que temas trata para decidir si realmente lo leeremos o no. Lo que uno haría sería leer el resumen que normalmente está en la parte posterior. Este texto nos diría cuales son los temas principales que se tocarán en él. La extensión de este texto determinará la cantidad de información que se nos es provista. Si el texto es de una extensión pequeña, sólo será nombrado el tema principal del libro. Sin embargo, si el texto tiene una extensión mayor más información respecto a los temas tocados en el libro se nos será entregada, pudiendo nombrarse temas secundarios.

Para abordar el problema, en esta memoria se propone un sistema de resúmenes semiautomáticos para un conjunto de datos en formato RDF, enfocándose en los conjuntos de datos provistos por DBpedia [1].

Se enfocará la solución propuesta en conjuntos de datos en formato RDF por varios motivos: RDF es un modelo estándar para el intercambio de datos en la web, el modelo es esencialmente un grafo dirigido por lo que tiene todas las ventajas y generalidades de aquella estructura de datos y permite una navegación rápida y eficiente de los datos.

Además, dentro de los conjuntos de datos RDF disponibles, este trabajo se enfocará en los conjuntos provistos por DBpedia por tres razones: Primero, contiene grandes cantidades de información. Segundo, por ser núcleo en la interconexión de datos en la web (*Linked Data*). Y tercero, por la correspondencia con Wikipedia.

1.2. Motivación

A partir de la solución de este problema, se podrán realizar acciones útiles para distintos ámbitos: La creación de conjuntos de datos de prueba y de análisis preliminar que sean consistentes y representen de manera fidedigna de lo que trata el conjunto de datos completo. También para que al publicarse datos de gran volumen se publiquen junto a un pequeño conjunto que muestre que es lo que representa. Por ejemplo, lo que hace DBpedia es publicar sus conjuntos de datos y pone a disposición un pequeñísimo conjunto de los primeros datos con los cuales busca mostrar el formato en que vienen. Esto no permite saber a priori, sin tener que obtener el conjunto completo de datos, si los datos sirven o no para el propósito que se les quiere.

1.3. Objetivo General

El objetivo del trabajo es analizar distintas soluciones para realizar resúmenes semiautomáticos, determinando su exactitud y sus falencias, para un conjunto de datos RDF. Además, implementar la mejor solución de las analizadas.

1.4. Objetivos Específicos

1. Analizar nociones de relevancia de trabajo relacionado.
2. Determinar el conjunto de datos donde trabajar y aplicar los métodos diseñados.
3. Diseñar e implementar un método que genere resúmenes semiautomáticos basado en arbolización y ranking más expansión.
4. Evaluar las soluciones propuestas.

1.5. Estructura de la memoria

El resto de la memoria se organiza de la siguiente manera:

El Capítulo 2 presenta los antecedentes, conceptos y definiciones que estarán presente durante todo este documento. Además, en el Capítulo 2 se presenta la revisión bibliográfica. En particular, se presenta lo que hay respecto al término de relevancia.

En el Capítulo 3 se describe en detalle el problema para el cual se presenta una solución.

Luego, en el Capítulo 4 se describen las alternativas propuestas y la descripción de la solución elegida. La descripción de la solución elegida se presenta junto al algoritmo y su descripción paso a paso.

En el Capítulo 5 se detalla el conjunto de datos escogido para la validación de la solución propuesta. Se detalla como se tomaron muestras para la experimentación preliminar y la experimentación formal, para que fueran representativas y no produjeran ningún sesgo en los resultados obtenidos. Luego, se describen los métodos de la experimentación, el por qué de cada uno y lo que se busca a través de ellos. Finalmente, se presentan los resultados obtenidos de la experimentación formal.

Por último, en el Capítulo 6 se presentan las conclusiones del trabajo realizado. Se comentan respecto a los objetivos planteados y respecto a los resultados obtenidos. Además, se analizan las posibles mejoras que podrían realizarse y los caminos a seguir.

Capítulo 2

Antecedentes

Este capítulo especifica temas y conceptos importantes para el contexto donde se desarrolló esta memoria. Se explica lo que es la web semántica, contexto elegido para lidiar con el problema; el framework RDF, formato de los datos con los que se trabajó; *Linked Data*, método para la publicación de datos en la web; DBpedia, comunidad con el objetivo de estructurar contenido de Wikipedia. Además, se presenta la revisión bibliográfica respecto a la noción de relevancia en la web semántica, describiendo los puntos importantes de cada trabajo y en qué se relaciona con el trabajo realizado.

2.1. Web Semántica

La web semántica es un movimiento colaborativo liderado por la W3C que promueve el uso de un formato común para la información en la web con el objetivo de que los datos puedan ser compartidos, reusados por aplicaciones y leídos y navegados por una máquina. Se basa en la idea de añadir metadatos semánticos y ontológicos a los datos de la web y a la web misma. Estos metadatos describen el contenido, el significado y la relación de los datos. De esta manera, se puede cumplir el objetivo principal de ser posible evaluarlos automáticamente por máquinas o agentes inteligentes.

"Si HTML y la web hicieran que todos los documentos online se vieran como un gran libro, RDF, esquemas y lenguajes de inferencia harán que todos los datos en el mundo se asemejaran a una gran base de datos." - Tim Berners-Lee

Tal como lo dice Tim Berners-Lee, considerado el padre de la web, si la web semántica se aplicara en su cabalidad en la web, podría considerarse y actuar sobre ella como si fuera una gran base de datos. La forma en que se añaden metadatos y describen las relaciones de los datos está construida sobre la especificación RDF.

2.1.1. RDF

El Marco de Descripción de Recursos (del inglés *Resource Description Framework*, RDF) es un framework para metadatos en la web, desarrollado por la W3C. Se encarga de describir la información y los recursos de la web. RDF se basa en la idea de convertir las descripciones de los recursos en expresiones del tipo sujeto-predicado-objeto, lo cual se denomina *triple*. El sujeto se refiere al recurso que estamos describiendo. La propiedad que se desea describir del recurso es el predicado. El objeto es valor de la propiedad u otro recurso con el que se establece la propiedad. Los sujetos pueden ser una URI o un nodo blanco, los predicados son URI y el objeto puede ser tanto como una URI como un literal, es decir, un string que representa un valor o un nodo blanco. Un nodo blanco es un recurso para el cual no fue asignada una URI.

Por ejemplo, si quisiéramos representar la frase "Alicia tiene 23 años" tendríamos el siguiente triple: El sujeto representado por "Alicia", el predicado representado por la propiedad "edad en años" y el objeto representado por el valor literal "23".

Un conjunto de triples RDF intrínsecamente representa o puede ser representado por un multigrafo dirigido y etiquetado. Los sujetos y objetos son representados por nodos y los predicados por arcos etiquetados.

Siguiendo el ejemplo anterior, agreguemos al conjunto las siguientes frases: "Alicia vive en Francia", "Alicia tiene dos amigos llamados Mauricio y Juan" y "Juan es amigo de Mauricio". De esta forma, se obtiene el siguiente grafo donde podemos observar todas las relaciones descritas de la forma nodo-arco-nodo. Además, podemos observar características implícitas. Por ejemplo, Alicia, Juan y Mauricio forman un grupo de amigos.

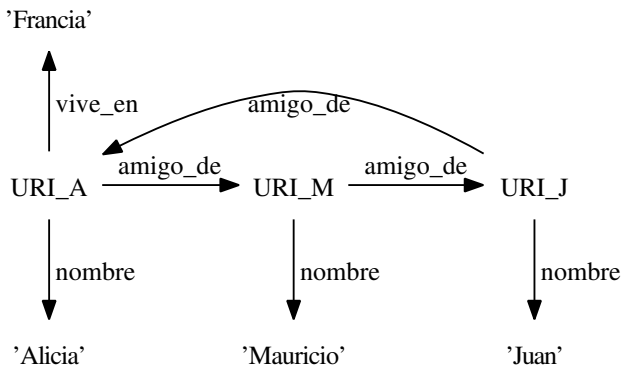

Figura 2.1: Grafo de los triples RDF.

2.2. Linked Data

Como se dijo anteriormente, la web se podría tratar como una gran base de datos gracias a la web semántica. Esto porque no se trata sólo de poner los datos a disposición, sino

de enlazarlos con otros gracias a los metadatos que añadimos. Esta interconexión de datos (*Linked data* en inglés) permite que podamos encontrar más datos relacionados a partir de uno.

Linked data tiene 4 principios básicos:

1. Utilizar URI como nombres para los recursos.
2. Utilizar URI HTTP para que la gente puede buscar esos nombres.
3. Cuando alguien busque una URI, proveerle información útil, usando los estándares (RDF, SPARQL)
4. Incluir enlaces a otras URI para que la gente pueda descubrir más cosas.

Dentro de los conjuntos de datos que encontramos en *Linked data*, DBpedia es uno de los más conocidos e importantes. Lo describiremos a continuación.

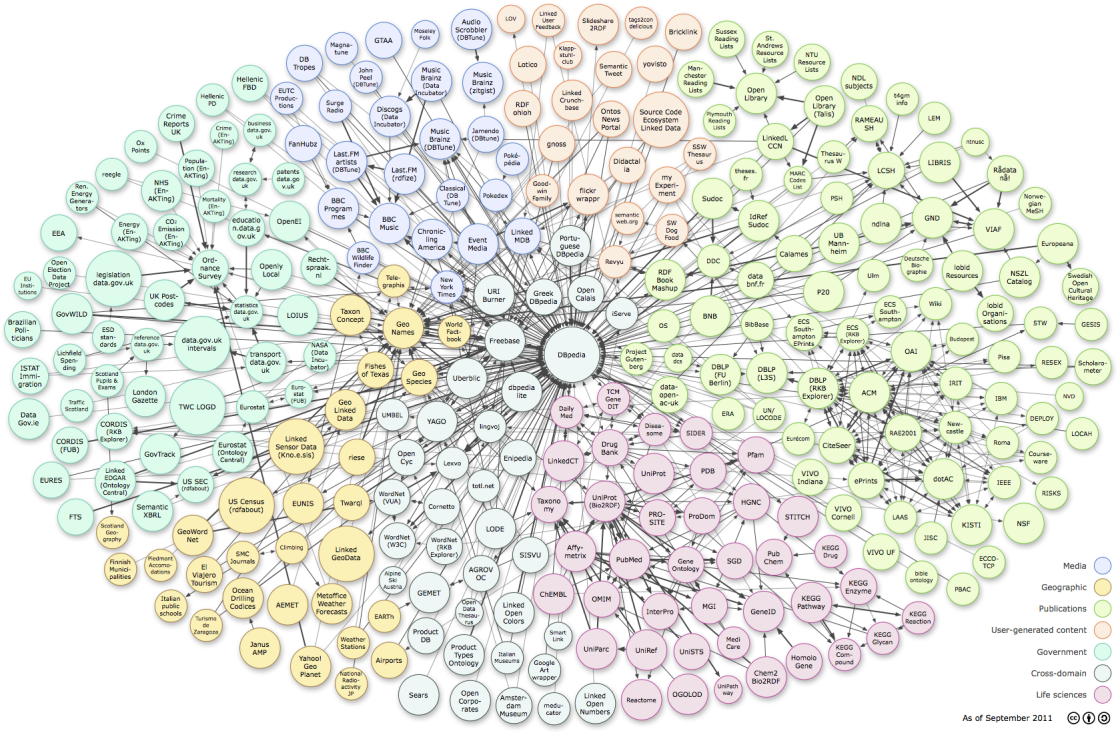

Figura 2.2: Diagrama de Linked Data - Septiembre 2011.

2.2.1. DBpedia

DBpedia es una comunidad cuyo objetivo es extraer información estructurada de Wikipedia y hacer esta información disponibles en la web. Esto con el objetivo de que la gran cantidad de información que contiene pueda ser usada en nuevas e interesantes formas, que pueda inspirar nuevos mecanismos, tanto de navegación como enlace, y mejorar la enciclopedia en sí.

La base de conocimiento de DBpedia actualmente describe más de 3.64 millones de elementos hasta en 97 idiomas. Un tercio de estos elementos están extraídos de la versión en inglés de Wikipedia. Entre estos elementos tenemos, por ejemplo: 416.000 personas, 526.000 lugares, 106.000 álbumes de música y 60.000 películas. El conjunto de datos también consiste en elementos no descritos por él, por ejemplo, 2.724.000 enlaces a imágenes, 6.300.000 enlaces a páginas externas y 740.000 categorías distintas.

En la comunidad de *Linked Data* de la W3C un gran número de entidades ha comenzado a publicar e interconectar datos en la web. El resultado de estos datos consiste actualmente en varios billones de triples RDF que cubren información de muchos tipos. Por ejemplo, información de personas, compañías, comunidades online, cine, música, libros, etc. DBpedia cumple un rol importante en este proyecto, ya que DBpedia define URI para millones de conceptos. Por este motivo muchos conjuntos de datos han empezado a enlazar sus conjuntos de datos con DBpedia, haciendo que DBpedia sea un punto central en la interconexión de datos en la web.

2.3. Trabajo Relacionado

Previo a la determinación de la propuesta de solución, se realizó una investigación sobre la noción de relevancia en la web semántica. Se realizó con el objetivo de encontrar que enfoques se utilizan para abordar el tema, ver qué avances hay en aquellos trabajos, ver qué caminos se han recorrido y cuáles serán los siguientes por recorrer.

De esta investigación previa cuatro trabajos fueron los más significativos.

2.3.1. Ranking de relaciones complejas

Este trabajo presenta SemRank[4], un modelo que se enfoca en rankear relaciones complejas entre dos recursos, llamadas asociaciones semánticas, según cuan predecibles serán para un usuario. Ya que es esperado que en la web semántica la cantidad de relaciones sea mucho mayor a la cantidad de entidades es válido pensar en un ranking para ellas. Además, se utiliza el argumento de las limitaciones de una visión centrada por entidades, citando a Grady Booch: "Un objeto por si solo es absolutamente poco interesante" [5].

Este trabajo rankea las asociaciones semánticas que existen entre dos recursos, digamos r_1 y r_2, según el tipo que formen. Se consideran tres tipos de asociaciones:

1. **ρ-pathAssociation:** Caminos directos entre r_1 y r_2.
2. **ρ-isoAssociation:** Caminos entre r_1 y r_n y entre r_2 y r_m donde los caminos sean semánticamente similares.
3. **ρ-joinAssociation:** Caminos entre r_1 y r_n y entre r_2 y r_n, es decir, que r_1 y r_2 se encuentren en algún punto.

Consideran que este ranking debe ser flexible, en el sentido de poder manejar el grado

de predictibilidad. Esto significa poder elegir qué nivel se desea utilizar para el ranking. Entre un modo convencional donde la respuesta son las asociaciones que tenderán a ser las más esperadas por el usuario, y un modo de descubrimiento donde, al contrario del modo convencional, se entreguen las asociaciones menos esperadas por el usuario.

El cálculo del valor de una asociación semántica en el ranking se compone de tres partes:

1. (I): Se utiliza teoría de la información para calcular que cantidad de información da cada asociación respecto al conjunto de datos para luego ser rankeada.
2. (RC): Refracción, lo cual se refiere a la creación de caminos no explícitos en el esquema de los datos.
3. (S-Match): Concordancia semántica, donde se evalúan las palabras claves entregadas en la query con las asociaciones semánticas.

Considerando estos tres puntos se obtiene la fórmula (de forma simplificada):

$$SemRank(ps) = I(ps) \cdot (1 + RC_{\mu}(ps)) \cdot (1 + SMatch(ps))$$

Las pruebas de SemRank se realizaron sobre datos sintéticos, es decir, datos compuestos por los autores de SemRank. El argumento para realizar las pruebas con aquellos datos fue que los conjuntos de datos disponibles del mundo real no son apropiados para las pruebas y comprobaciones del nuevo ranking, ya que no hay una gran cantidad de relaciones distintas en ellos, no hay variedad, simplemente se repiten unas pocas.

Los resultados de las pruebas mostraron que el ranking entrega resultados de la forma esperada. Aunque parece ser que fueron realizadas sólo unas pocas pruebas.

Con este trabajo se podrían determinar las relaciones más importantes dentro de un conjunto de datos. Por ello, los nodos que forman estas relaciones también tienen cierto nivel de importancia. Por otro lado, podrían determinarse cuáles son los nodos que forman parte de las relaciones más importantes con mayor frecuencia y así tener mayor información.

2.3.2. Método para rankear nodos en un grafo RDF

Este trabajo presenta *noc-order* (*NOde Centrality Ordering*) [12], una métrica para rankear nodos RDF basada en la noción de centralidad.

Calcular *noc-order* es equivalente a calcular el camino más corto entre todos los pares de nodos en grafo. Para ello, se necesita especificar una métrica de distancia.

Para la distancia se utilizan tres definiciones principales. La primera de ellas es $\mathrm{d}_o(x, y, z) = m$, donde m es el número de arcos en el camino p (definición usual de distancia). La segunda es $\mathrm{d}_w(x, y, z) = \sum_{\mathrm{i}=1}^{m} w(\mathrm{e})$. Esta es la suma de todos los pesos de los arcos en el camino basado en una función de peso w, tal que $w : E \to \Re$. En este trabajo se considera la frecuencia de ocurrencia de cada predicado como función de peso. Entre más común es un predicado, menos relevante es para el ranking presentado. Finalmente, definen $\mathrm{d}_w^{\alpha}(x, y, p) = \sum_{\mathrm{i}=1}^{m}(w(\mathrm{e})/\alpha^{\mathrm{i}})$ donde la distancia se vuelve más grande en función de α donde $0 \leq \alpha \leq 1$, es decir, caminos

cortos son más deseables que largos. Ya definidas estas distancias, se utilizan para el cálculo de los caminos más cortos entre todos los pares de nodos y así calcular *noc-order*.

Este trabajo [12] se relaciona directamente con el problema que se desea resolver, ya que propone una métrica de ranking de nodos. Se basa en centralidad lo que denota un nivel de importancia en los nodos.

2.3.3. Ranking de conocimeinto en la Web Semántica

Este trabajo [8] busca ayudar a los agentes a navegar la web semántica y rankear resultados de búsqueda. Expone que los modelos de navegación y ranking actuales no son adecuados para la web semántica por dos razones. Primero, no diferencian documentos web semánticos de la gran cantidad de otras páginas web. Segundo, no parsean y no usan la estructura interna de un documento semántico y enlaces semánticos externos entre ellos. Para ejemplificar esto, muestran que Google, uno de los mejores motores de búsqueda web de la actualidad, tiene un rendimiento pobre si se trata de buscar ontologías. Buscan "person ontology" y la ontología FOAF (la más utilizada para describir una persona) no aparece dentro de los 10 primeros resultados.

Este trabajo [8] se enfoca en rankear ontologías en distintos niveles para promover la reutilización de ellas. Los ranking de ontologías a nivel de documento han sido ampliamente estudiados, ya que la mayoría son publicadas a través de documentos semánticos que las definen. Por ello se busca otra forma para realizar el ranking.

Se considera a la web semántica materializada en la web. Y para navegarla, un usuario no puede confiar en la semántica de las referencias de URI por tres razones. Primero, el namespace de una referencia apunta al documento que la define pero no de vuelta. En segundo lugar, rdfs:seeAlso ha sido usado ampliamente para interconectar documentos semánticos en aplicaciones basadas en la ontología FOAF, pero rara vez funciona en otros documentos semánticos. Finalmente, en tercer lugar, owl:imports no interconecta ontologías, pero aquellas relaciones son raras, ya que las ontologías normalmente son desarrolladas y distribuidas independientemente. Por estas razones les es válido pensar en un nuevo modelo de navegación.

Debido a que normalmente los grafos RDF son accedidos en un nivel de documentos, se ha simplificado el modelo de navegación generalizando los caminos de navegación a tres niveles.

1. (EX) Extensión. Cuando un documento define un término utilizando términos definidos en otro.
2. (TM) Uso de término. Relación entre dos documentos cuando uno usa un término definido en otro.
3. (IM) Importar. Cuando un documento importa directamente o transitivamente otro documento.

Con ello se propone *OntoRank*, un ranking a nivel de documento basado en el modelo surfer que emula a un agente navegando a nivel de documento. En él puede seguir un enlace a otro documento o saltar de forma aleatoria a un nuevo documento con una probabilidad

constante de $1 - \mathrm{d}$. Sea $link(\alpha, l, \beta)$ un enlace semántico entre un documento α a uno β con un tag l; $linkto(\alpha)$ un conjunto de documentos que tienen un enlace directo al documento α; $weight(l)$ es el tipo de preferencia en la navegación de enlaces; $OTC(\alpha)$ un conjunto de documentos que importan (IM) o extienden (EX) α como una ontología. A continuación se define la fórmula para calcular *OntoRank*.

$$wPR(\alpha) = (1 - \mathrm{d}) + \mathrm{d} \sum_{x \in linkto(\alpha)} \frac{wPR(x) \cdot f(x, \alpha)}{\sum_{link(x,\_,\_y)} f(x, y)}$$

$$f(x, \alpha) = \sum_{link(x,l,\alpha)} weigth(l)$$

$$OntoRank(\alpha) = wPR(\alpha) + \sum_{x \in OTC(\alpha)} wPR(x)$$

Los autores de *OntoRank* realizaron pruebas sobre un conjunto de datos reales DS-APRIL recolectado por Swoogle en abril de 2005. Este conjunto de datos contenía más de 300 mil documentos. 1,5 % en ontologías, 24 % en documentos FOAF y 60 % en documentos RSS. Los resultados fueron un 40 % de mejora en las búsquedas de ontologías con *OntoRank* sobre PageRank.

Este trabajo es interesante, ya que propone un método para rankear ontologías. Podría utilizarse de modo indirecto en el problema que se desea resolver. Por ejemplo, en el trabajo presentado en la sección 2.3.2 definen *noc-order* y utilizan la frecuencia de predicados como parte de su método.

2.3.4. Eliminación de conflictos en Web Semántica

Este trabajo[7] muestra los conflictos semánticos que ocurren en la minería web y propone la utilización de métodos de extensión para eliminar los distintos tipos de conflictos que un agente reporte. Con ello busca mejorar las búsquedas realizadas en la web semántica y mejorando con ello la recuperación de la información.

Los conflictos que pueden producirse son desajustes en la ontología, entre los que se incluyen los siguientes:

1. Mismos términos para diferentes conceptos.
2. Diferentes términos para mismos conceptos.
3. Atributos semánticamente similares que tienen diferentes significados en sus dominios.
4. Atributos que tienen diferente generalización y niveles de agregación.
5. Mismos atributos, pero que tienen diferentes requerimientos de calidad de información.

La metodología propuesta es: (1) analizar qué tipo de conflicto ocurre, (2) si es necesario, representar objetos para distintos significados con elementos básicos, (3) escoger el método de extensión adecuado para eliminar el conflicto.

Por ejemplo, en una página se podría encontrar la frase "Yo uso mi computador para navegar páginas web" y en otra la frase "Yo ocupo mi máquina de escritorio para navegar páginas web". Aquí es donde el agente reportaría un conflicto. Se identificaría que el conflicto es que se utiliza "computador" y "máquina de escritorio" para el mismo concepto. Luego, se procedería a reemplazar ambos términos por un objeto genérico. El siguiente paso sería escoger el método adecuado para eliminar el conflicto. En este ejemplo correspondería a un método de árbol divergente. Básicamente, se refiere a que si dos objetos semánticos tienen las mismas características y medidas correspondientes, entonces son el mismo objeto. Con ello se comprueba que "computador" y "máquina de escritorio" son el mismo objeto.

Este trabajo [7] simplemente plantea la idea de utilizar métodos para eliminar conflictos semánticos, ya que no propone métodos para cada conflicto mostrado y no hay implementación ni pruebas reales para comprobar correctitud y rendimiento. No obstante, son interesantes los conflictos que se plantean, pues dentro de la determinación de recursos importantes podrían ser motivos de deterioro de resultados en la resolución del problema a resolver.

2.3.5. Ranking de similitud

Este trabajo define SimRank[13], una métrica de similitud. Es aplicable en cualquier conjunto donde existan relaciones de objeto a objeto. Mide la similitud del contexto estructural, basado en sus relaciones con otros objetos. En palabras simples, SimRank declara que 'dos objetos son similares si están relacionados con objetos similares'.

Para un nodo v, se denota por $I(v)$ y $O(v)$ al conjunto de vecinos que tienen arcos incidentes a v y al conjunto de vecinos que v incide en ellos respectivamente.

La similitud entre dos objetos a y b se define como $s(a,b) \in [0,1]$. Si $a = b$ entonces $s(a,b)$ está definido como 1. En caso contrario se define como:

$$s(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{\mathrm{i}=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_{\mathrm{i}}(a), I_{\mathrm{i}}(b))$$

Los autores de SimRank realizaron pruebas experimentales sobre dos conjuntos de datos. El primero, un conjunto de papers de investigación científica de $ResearchIndex$. Contiene 688.898 referencias entre 278.628 papers. El segundo, es un conjunto de notas académicas de 1.030 alumnos de la escuela de ingeniería de Stanford. Cada nota contiene la lista de todos los cursos que ha cursado el alumno; un promedio de 40 cursos por estudiante. Mostró buenos resultados, considerablemente mejores comparados con una métrica de co-citación simple.

Gracias a que SimRank determina el nivel de similitud se podrían generar grupos de objetos similares para facilitar la determinación de resúmenes; se podría disminuir la cantidad elementos con los que se trabaja; o simplemente se podrían encontrar otros objetos dentro del conjunto de datos si es que previamente se determinó uno importante con otros métodos.

Capítulo 3

Descripción del Problema

En la actualidad, la cantidad de información que se genera en el mundo es inmensa. En el campo científico tenemos, por ejemplo, al campo de la astronomía, donde un telescopio puede producir unos 60TB de información cada semana. También tenemos al campo de la biología con cadenas de ADN o aminoácidos. Incluso algo tan simple como el pronóstico del clima puede producir grandes cantidades de información en el campo de la meteorología. No sólo en el mundo científico se produce este fenómeno, por ejemplo, un usuario navegando por internet produce grandes cantidades de información: Comentarios en foros, la participación en redes sociales o simplemente la comunicación a través de la red.

Consideremos cuatro conceptos para establecer el contexto donde se desarrollará el problema: Primero, la web, un espacio para la compartición de datos. Ésta se ve afectada directamente con esta situación. Al ser una gran red de información intercambiable, el crecimiento afecta la navegación a través de ella. Segundo, *Linked Data*, método que describe como publicar datos para que así puedan ser enlazados por otros y sean más útiles. Tercero, la web semántica, movimiento liderado por la W3C, promueve el uso de un formato común para la información en la web con el objetivo de que los datos puedan ser compartidos, reusados por aplicaciones y leídos y navegados por una máquina. Cuarto, el modelo RDF, modelo estándar propuesto por la W3C para el intercambio de datos que representa la información de los recursos en la web. La unión de estos cuatro conceptos establece el contexto donde se verá el problema: Conjuntos de datos RDF.

Para conjuntos de datos de gran volumen aparecen problemas de almacenamiento y procesamiento. En particular, si uno necesita o encuentra un conjunto de datos donde realizar un análisis de cierto tipo. A priori, no se sabe si el conjunto sirve para lo que se desea. Puede que sea de temas que no interesan o que no se ajustan a los que buscamos. Frente a esta situación, nos planteamos la siguientes preguntas: ¿Qué se puede hacer? ¿Cómo manejarlos? ¿Cómo sé si sirve para los análisis que deseo hacer? Estas preguntas podrían responderse si se tuviera un resumen del conjunto de datos. Por ello, esta memoria se enfoca en la realización de resúmenes de conjuntos de datos RDF. Dado el gran volumen de los conjuntos, se proyecta realizar los resúmenes de forma semiautomática.

La resolución del problema se centrará en los conjuntos de datos provistos por DBpedia.

Se eligió DBpedia por tres razones: Primero, contiene grandes cantidades de información. Describe más de 3.64 millones de elementos en más de 90 idiomas. Segundo, es núcleo en la interconexión de datos en la web (*Linked Data*). Esto da la oportunidad de poder navegar muchos otros datos desde DBpedia, extendiendo el estudio hacia otros conjuntos de datos eventualmente. Tercero, por su correspondencia con Wikipedia. Esto permite realizar comparaciones de métricas, resultados y otros estudios, entre DBpedia y Wikipedia, es decir, comparar la versión estructurada con la no estructurada de un conjunto de datos.

La solución de este problema será de gran utilidad para distintos ámbitos: La creación de conjuntos de datos de prueba y de análisis preliminar. También para que al publicarse datos de gran volumen se publiquen junto a un pequeño conjunto que muestre que es lo representa. Por ejemplo, lo que hace DBpedia es publicar sus conjuntos de datos y pone a disposición un pequeñísimo conjunto de los primeros datos con los cuales busca mostrar el formato en que vienen. Esto no permite saber a priori, sin tener que obtener y analizar el conjunto completo de datos, si los datos sirven o no para el propósito que se les quiere.

Capítulo 4

Descripción de la Solución

En este capítulo se detalla la solución propuesta. Primero, se explica la solución propuesta para resúmenes semiautomáticos. Se detallan los puntos más importantes para comprenderla. Junto a ello, se muestra el algoritmo asociado a la solución y se describe paso a paso. Luego, se describen temas como la complejidad del algoritmo y temas de implementación de la solución. Finalmente, se presentan las alternativas de solución que se consideraron en un inicio, pero que se descartaron por distintas razones. Se explica por qué se consideraron en su momento y qué es lo que se tomó de ellas para la solución propuesta.

4.1. Solución Propuesta

La solución que se propone es una solución que derivó de las dos alternativas que se tenían. La idea base de esta solución propuesta es poder encontrar individualmente en el grafo que conforma el conjunto de datos, los puntos más relevantes e importantes global y localmente. A partir de estos puntos, se podrá ir generando el resumen del conjunto de datos.

El conjunto de datos que se escogió para diseñar e implementar la solución fue DBpedia (ver 2.2.1). Se escogió por cuatro razones: Por ser un gran conjunto de datos de distintos tópicos, por tener el formato adecuado, por ser un conjunto importante dentro de la Linked Data y por su correspondencia con Wikipedia.

La solución trata al grafo como un grafo no dirigido y rankea los nodos del grafo según dos métricas. Luego, elegimos cierta cantidad de nodos que sean los mejores evaluados de cada métrica. Esta selección de nodos es la que se retorna como resultado de la solución.

Se toma el grafo correspondiente al conjunto de datos como un grafo no dirigido, porque se está más interesado en la existencia de una relación entre dos recursos que en la dirección de la relación. Además, siempre para un predicado podremos considerar su inverso. Por ejemplo, si tenemos el predicado `padreDe` podemos tener `hijoDe`.

Las dos métricas utilizadas sobre los nodos del grafo para poder determinar los puntos

más importantes dentro del conjunto de datos fueron:

- Centralidad de intermediación (*Betweeness centrality* en inglés).
- Grado de entrada y salida.

La centralidad de intermediación es una medida de centralidad que es presentada como una métrica para cuantificar la responsabilidad dentro de una red. Por ello, un vértice tiene una alta centralidad de intermediación cuando tiene una alta probabilidad de aparecer en el camino más corto entre dos vértices elegidos uniformemente al azar. Gracias a esta métrica podemos determinar en forma global los recursos más importantes.

Por otra parte, tenemos los grados de los vértices. Gracias a esta medida podemos determinar en forma local los recursos más importantes. El que un vértice tenga alto grado quiere decir que referencia a muchos o que es altamente referenciado. Ésto indica que aquel vértice tiene muchas propiedades o que es parte de propiedades de muchos otros recursos. En primera instancia, en vez de la utilización de los grados, se planeó la utilización de PageRank. PageRank se basa en la referencia de los enlaces y, ya que se considera al grafo no dirigido, perdía sentido su utilización. En cambio, la distribución de nodos tiene más sentido, ya que le es indiferente la dirección. Además, en un grafo no dirigido se tiene una alta correlación entre los grados y PageRank de cada nodo.

En un algoritmo, un tema importante es su complejidad. En esta solución hay cuatro puntos importantes donde se debe determinar la complejidad. Consideremos un grafo $G = (V, E)$. Primero, está la determinación del grados de cada nodo. Determinar los grados de los nodos se considera de orden $O(V^2)$ (V cantidad de vértices en el grafo) en el peor caso, dependiendo de cómo se tiene representado el conjunto, si en una lista o matriz de adyacencia. Segundo, tenemos la acción de transformar el grafo a un grafo no dirigido. Ello es orden $O(V^2)$ en el peor caso, ya que también depende de como se tenga representado el grafo. Tercero, ordenar una lista de nodos se puede realizar en tiempo $O(Vlog(V))$. Finalmente, tenemos el cálculo de la centralidad de intermediación de los nodos. Este cálculo es el más complejo, ya que se necesita calcular los caminos más cortos entre todos los pares de nodos. Se puede realizar con el algoritmo de Floyd-Warshall en $O(V^3)$ y, si el grafo es poco denso, se puede disminuir a $O(VE)$ con el algoritmo de Brandes[6] (E cantidad de arcos en el grafo). Tomando estos cuatro puntos en consideración, se determinó que el algoritmo de la solución es de orden $O(V^3)$.

A continuación se presenta el algoritmo de la solución, luego se procede a explicar paso a paso cada parte:

4.2. Algoritmo

input : Un grafo RDF G y un factor $r \in [0, 1]$
output: Conjunto de nodos S

1 $G \leftarrow$ `Limpieza`(G);
2 $G \leftarrow$ `NoDirigido`(G);
3 $ListCent \leftarrow$ `ListaCentralidad`(G);
4 $ListGrad \leftarrow$ `ListaGrados`(G);
5 $S \leftarrow$ `ExtraerPrimerosDeListas`$(ListCent, ListGrad, r)$;
6 **return** S;

Algoritmo 1: Algoritmo de resúmenes semiautomáticos

4.3. Descripción del Algoritmo

El algoritmo recibe como input dos argumentos. Primero un grafo G y, segundo, un factor $r \in [0, 1]$, el cual es el factor de los nodos totales que se retornará. Como output entrega un conjunto de vértices que representan el resumen del grafo entregado. El tamaño de este conjunto es, como ya se dijo, el factor r de la cantidad de nodos del grafo G.

A continuación se detallarán los cinco métodos principales.

1. `Limpieza`: Se realizó este procedimiento para eliminar características que no aportan al algoritmo ni a los resultados y sólo sesgan el resultado. Tenemos dos procedimientos dentro de limpieza. El primero consiste en quitar todos los nodos etiqueta del grafo. Los quitamos porque no influyen dentro del análisis, simplemente lo degeneran. Para cada etiqueta hay un sólo nodo que lo referencia, el cuál es su propio recurso. En segundo lugar, se eliminaron ciertos nodos que eran referenciados por todo el resto de los nodos. Por ejemplo, en el caso del conjunto de datos de las categorías, todos los recursos son del tipo concepto $<$ *http://www.w3.org/2004/02/skos/core#Concept* $>$. Esto no hace ninguna diferencia dentro del conjunto de datos, ya que es una característica redundante si sólo trabajamos dentro aquel conjunto.
2. `NoDirigido`: Esto simplemente transforma el grafo a un grafo no dirigido. Como se dijo anteriormente, nos interesan las relaciones que existen más que la dirección. Además, para cada predicado entre dos recursos, siempre es posible tener el predicado inverso (por ejemplo, `publicó` el inverso de `publicadoPor`).
3. `ListaCentralidad`. Este procedimiento calcula la centralidad de intermediación para para cada nodo en el grafo. Luego, ordena los nodos según el valor calculado de mayor a menor. Finalmente, entrega una lista de los nodos ordenados.
4. `ListaGrados`. Este procedimiento es similar a `ListaCentralidad`. Para cada nodo se calcula su grado. Se ordenan los nodos según el grado calculado, de mayor a menor. Finalmente, se entrega una lista de los nodos ordenados.
5. `ExtraerPrimerosDeListas` se encarga de seleccionar los nodos de ambas listas que serán retornados. El método toma ambas listas, las retornadas por `ListaCentralidad`

y `ListaGrados`, y el factor r. Se calcula la cantidad de nodos que se extraerán de las listas. Si n es la cantidad de nodos en el grafo, entonces rn nodos serán extraídos. La extracción se divide en dos partes. Primero, se toman los primeros 0,8rn nodos de la lista *ListaCent* y se guardan en un conjunto S. Así tendremos los nodos con mayor centralidad de intermediación. Y segundo, se toman los 0,2rn primeros nodos de la lista *ListaGrad* que no pertenezcan al conjunto S, es decir, que no se encuentren en los 0,8rn primeros nodos de la lista *ListGrad* y se unen al conjunto S.

Se definió un factor de 0,8 en la elección de nodos de la lista *ListaCent* y 0,2 para la lista *ListaGrad* porque se desea dar más importancia a los nodos elegidos con centralidad de intermediación que a los elegidos con grados. Además, porque durante el desarrollo del método se observó que tomando mayor cantidad de nodos de la lista *ListaGrad*, se tomaban nodos con bajo grado, lo cuál no aportaba al resultado de la solución sino que al contrario.

6. Finalmente, se retorna el conjunto S.

4.4. Implementación

La solución propuesta se implementó utilizando el lenguaje *Python 2.7*, utilizando el paquete *igraph 0.6* [1] para el manejo de grafos. Se utilizó este lenguaje, ya que sus características de ser interpretado y dinámicamente tipado ayudó al manejo de los conjuntos de datos. Por ejemplo, sólo debía cargarse el conjunto de datos en memoria una sola vez y no cada vez que se ejecutaban los experimentos. Además, el paquete *igraph* está implementado para *Python*. El paquete *igraph* ayudó en el manejo de los grafos con algoritmos ya implementados de forma eficiente y probados.

El conjunto de datos se recibió en formato *N-Triples* y se transformó a formato *Graph Modelling Language* (GML) para su manejo. *N-Triples* es un formato para la transmisión de la información donde cada línea representa una afirmación. Cada afirmación consta de tres partes separadas por un espacio: Sujeto, predicado y objeto. Cada una es terminado por un punto.

4.5. Alternativas descartadas de solución

Antes de llegar a la solución propuesta y luego de haber investigado sobre la noción de relevancia en la web semántica, se llegaron a dos alternativas para la propuesta de una solución que se descartaron por distintos temas. Las alternativas descartadas de solución se formularon analizando el trabajo relacionado. De él se observó que las ideas más recurrentes eran sobre: Similitud ente nodos, la forma de los caminos entre recursos, concepto de centralidad sobre nodos y eliminación de 'ruido' en las búsquedas o recuperación de información. La formulación de alternativas se realizó apuntando a dos de estos temas: Idea de centralidad y similitud

[1] `http://igraph.sourceforge.net`

entre nodos. Se incluyen en este informe estas alternativas, pues se considera que pueden ser de utilidad para futuros investigadores de esta área.

4.5.1. Arbolización

La representación de un grafo (dirigido o no) como árbol tiene ciertas propiedades que facilitan su análisis. Por ejemplo, la asociación de distintos vértices del grafo en un nodo del árbol o la simplificación de caminos, quedando un camino único para dos pares de nodos. Por ello se considera utilizar arbolización para facilitar los análisis que quieran hacer sobre los datos.

En particular, para este caso se planea utilizar un algoritmo de arbolización (*Tree decomposition* en inglés) sobre el grafo que forma el conjunto de datos y así determinar la relevancia de los nodos por los niveles del árbol. Entre más cercano a la raíz, más relevante y/o representativo. Se planea evaluar heurísticas para cota superior e inferior además de un algoritmo *complete anytime* [11] [15].

Esta alternativa se dejó de lado por temas de rendimiento y escalabilidad, ya que con las herramientas disponibles para la arbolización de un grafo, a lo más un grafo de unos 5000 nodos podía ser procesado en un tiempo razonable (menos de una hora). Sin embargo, se observó una alta correlación de los nodos en el grafo y los nodos correspondientes al árbol generado respecto a métricas como la centralidad de intermediación y grados de un nodo.

4.5.2. Ranking y expansión

Otro enfoque para este problema es poder determinar ciertas partes puntuales del conjunto de datos que son importantes y expandirlas tratando de conectarlas. Tomando nuevamente el ejemplo de un libro, si determinamos que los temas A y B son importantes, los temas que conecten A y B también lo serán ya que ayudan a dar un mejor significado a A y B. Para este caso se planea utilizar un tipo de ranking sobre los nodos del grafo: *noc-order* [12] y, luego, a partir de los nodos mejor rankeados expandir el grafo a través de nodos vecinos que tengan mayor similitud entre ellos determinado por distintas métricas: SimRank [13] y SemRank [4]

Esta alternativa se descartó porque se enfocaba más en determinar la similitud entre nodos más que en determinar los nodos importantes de un inicio. Sin embargo, se consideró para la solución propuesta el seleccionar nodos a través de una métrica de centralidad.

Capítulo 5

Conjunto de datos - DBpedia

Este capítulo trata sobre los conjuntos de datos que se utilizaron en esta memoria, ya sea considerados para su utilización, analizados o empleados para el diseño y desarrollo de la solución. Primero se describirán los conjuntos que provee DBpedia. Luego, se detallará cual fue el conjunto elegido para el diseño, desarrollo e implementación de la solución y por qué. Finalmente, se mostrará un análisis más en detalle del conjunto de datos elegido.

5.1. Conjuntos disponibles

Para poder realizar el diseño y desarrollo de la solución, se debía contar con un conjunto de datos desde un inicio. De esta manera, conociendo los datos desde el principio se podía responder mejor frente a resultados inesperados o simplemente poder dirigir mejor hacia donde llevar la solución. Además, se debían analizar para la posterior validación de la solución.

DBpedia tiene disponibles muchos conjuntos de datos. Por ejemplo, títulos de los artículos, resúmenes (*abstract*) de los artículos, títulos de las categorías de Wikipedia, redirecciones de algunos artículos, entre otros. De forma preliminar se eligieron tres conjuntos para ser analizados y de ellos seleccionar el que será utilizado en esta memoria. Estos tres conjuntos fueron elegidos porque no representan información uno a uno entre los recursos. Por ejemplo, los títulos de las categorías, donde cada artículo tiene sólo un título. También fueron elegidos porque representaban datos que un usuario pudiera comprender, es decir, que no fueran datos duros. Por ejemplo, los números identificadores de cada página.

Estas son las tres opciones consideradas:

- Enlaces entre artículos.
- Relación entre categorías.
- Categorías de los artículos.

A continuación se analizará brevemente cada una de las opciones.

5.1.1. Enlaces entre artículos

Este conjunto de datos describe los enlaces entre los artículos de Wikipedia. Por cada enlace que exista en un artículo de Wikipedia a otro artículo de Wikipedia, existirá un triple que describa esta relación. Por ello, el conjunto sólo contendrá recursos de DBpedia y ninguno externo.

Los triples tienen la siguiente forma:

$$< \texttt{artículo}_1 >, < \texttt{http://dbpedia.org/ontology/wikiPageWikiLink} >, < \texttt{artículo}_2 >$$

Dentro de los triples, el predicado utilizado siempre es el mismo. Los artículos involucrados son los que van cambiando. El conjunto de datos está conformado por 146 millones de triples. El grafo que conforma este conjunto tiene 16.719.396 nodos y 118.039.661 arcos.

5.1.2. Relaciones entre categorías

Wikipedia permite almacenar artículos y otras páginas en categorías. Estas categorías reúnen páginas de similares características. Las categorías tienen a su vez subcategorías, las cuales son más específicas, y supercategorías, que más generales. Permiten navegar de temas más generales a temas más concretos y viceversa, a través de una estructura de árbol. Además, estas categorías tienen categorías relacionadas. Las categorías relacionadas no denotan una supercategoría ni subcategoría, simplemente denotan a una categoría con la que se tiene algún vínculo, como por ejemplo, un tema en común.

Este conjunto de datos describe las relaciones entre las categorías de Wikipedia. Éstas son: La jerarquía de ellas, detallando las supercategorías para cada una; las categorías relacionadas entre sí, el tipo de cada recurso y la etiqueta de cada categoría. Los triples tienen la siguiente forma:

$$< \texttt{categoría} >, < \texttt{predicado} >, < \texttt{objeto} >$$

El "objeto" en los triples es otra categoría o $< \texttt{http://www.w3.org/2004/02/skos/core\#Concept} >$ para indicar que el tipo de una categoría es un concepto.

Dentro de los triples, tenemos la utilización de 4 predicados distintos. Se muestran a continuación junto al número de utilización de cada uno:

1. `http://www.w3.org/2004/02/skos/core#broader` - 1.463.237 veces.
2. `http://www.w3.org/1999/02/22-rdf-syntax-ns#type` - 740.362 veces.
3. `http://www.w3.org/2004/02/skos/core#prefLabel` - 740.362 veces.
4. `http://www.w3.org/2004/02/skos/core#related` - 28.748 veces.

Este conjunto se compone de 3.500.000 triples aproximadamente. El grafo de este conjunto de datos tiene 1.483.763 nodos y 2.972.709 arcos.

5.1.3. Categorías de los artículos

Como se explicó en 5.1.2, los artículos se pueden categorizar para su mejor navegación. Todo artículo de Wikipedia debe pertenecer por lo menos a una categoría. Gracias a esto, no tendremos nodos aislados que no referencien a ningún nodo o no sean referenciados por ninguno.

Este conjunto de datos describe las categorías que tiene cada artículo de Wikipedia. Cada triple describe un artículo que pertenece a una categoría. Los triples tienen la siguiente forma:

$$< \texttt{artículo} >, < \texttt{http://purl.org/dc/terms/subject} >, < \texttt{categoría} >$$

En los triples, el predicado siempre es el mismo. Los artículos y categorías involucradas son los que van cambiando en cada triple. El conjunto de datos está conformado por 13,6 millones de triples y al grafo correspondiente tiene 4.192.499 nodos y 13.610.094 arcos.

5.2. Conjunto de datos elegido

El conjunto elegido para el desarrollo de esta memoria fue el conjunto de relaciones entre las categorías.

Se eligió este conjunto por la combinación de varios puntos. Primero, se descartó utilizar el conjunto de datos de enlaces entre artículos por una razón: Los predicados que aparecen en los triples eran siempre los mismos. Al ser siempre el mismo predicado no entrega información semántica y sólo se trabaja con la estructura. Segundo, se descartó la utilización del conjunto de categorías de los artículos porque el conjunto de datos tenía demasiadas componentes conexas. Cada componente se constituía por el artículo y sus categorías. Esto no dejaba espacio a estructuras más complejas o relaciones entre todo el conjunto de datos. Con este conjunto no podría desarrollarse una buena solución. En cambio, el conjunto de relaciones entre los artículos muestra una estructura lo suficientemente compleja como para realizar con él la solución. Presenta cuatro predicados distintos, no así el resto de los conjuntos, con lo que tenemos mayor información semántica con la que trabajar. En definitiva, el conjunto de las relaciones de las categorías presenta lo que necesitamos para el diseño y desarrollo de la solución.

5.2.1. Características

En 5.1.2 se indicó que el conjunto contiene 1.483.763 nodos. De ellos 740.362 corresponden a etiquetas, 740.362 a URI de categorías y un nodo que representa concepto < `http://www.w3.org/2004/02/skos/core#Concept` >. Las relaciones entre estos nodos están representados por 2.972.709 arcos. Con estos arcos se representan las relaciones de supercategorías, categorías relacionadas, etiquetas correspondiente y tipo del recurso. En el grafo hay 740.362 arcos representando la relación etiqueta-categoría, mismsa cantidad de nodos

etiquetas, hay 1.463.237 arcos que representan la relación categoría-supercategoría, 28.748 arcos representan las categorías relacionadas y 740.362 representan el tipo de los recursos.

En la siguiente figura se muestra la distribución de grados del grafo en escala logarítmica. Se grafica la cantidad de nodos respecto a los grados, los grados de entrada y los de salida. Claramente, forman una ley de potencia.

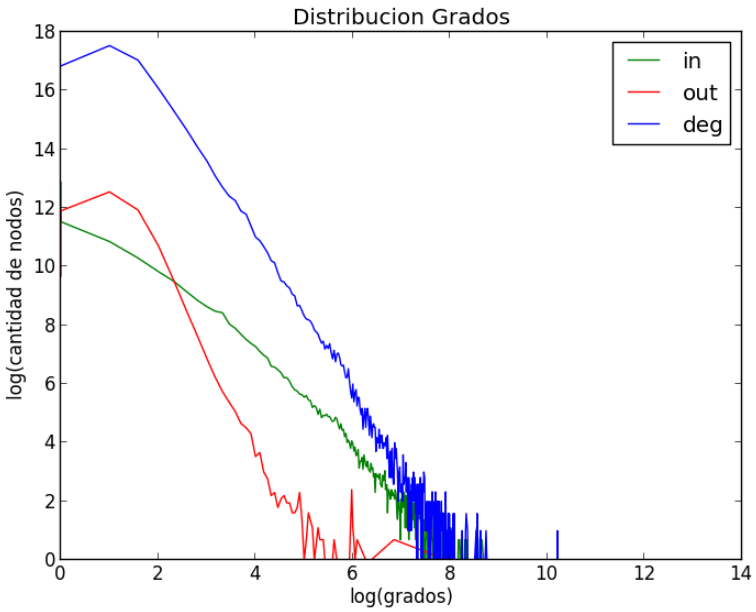

Figura 5.1: Distribución de grados del conjunto de datos

Para poder observar la estructura que tiene el grafo, se realizó búsqueda por amplitud desde un nodo elegido uniformemente al azar. En la figura 5.3, se puede observar el grafo que se generó desde la categoría 'pastos' (*Grasses*). Y en la figura 5.4, se puede observar el grafo generado desde la categoría 'Artículos de Dakota del sur según importancia' (*South Dakota articles by importance*). Para simplificar la figura, no se graficaron los nodos etiquetas.

Luego, para poder caracterizar más el conjunto de datos, se tomaron muestras de la misma manera que antes, calificando los nodos según centralidad de intermediación y PageRank. A cada nodo se le asignó un color para cada índice normalizado entre 1 y 10, siendo 1 el más importante y 10 el menos importante (ver figura 5.2). PageRank se asignó en el color de borde y la centralidad de intermediación en en interior. En las figuras 5.5 y 5.6 se pueden observar los grafos generados desde los nodos 'ABBA' (el grupo musical) y desde 'los álbumes de Karl Berger' (*Karl Berger Albums*).

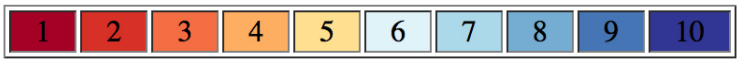

Figura 5.2: Colores para la escala de centralidad de intermediación y PageRank

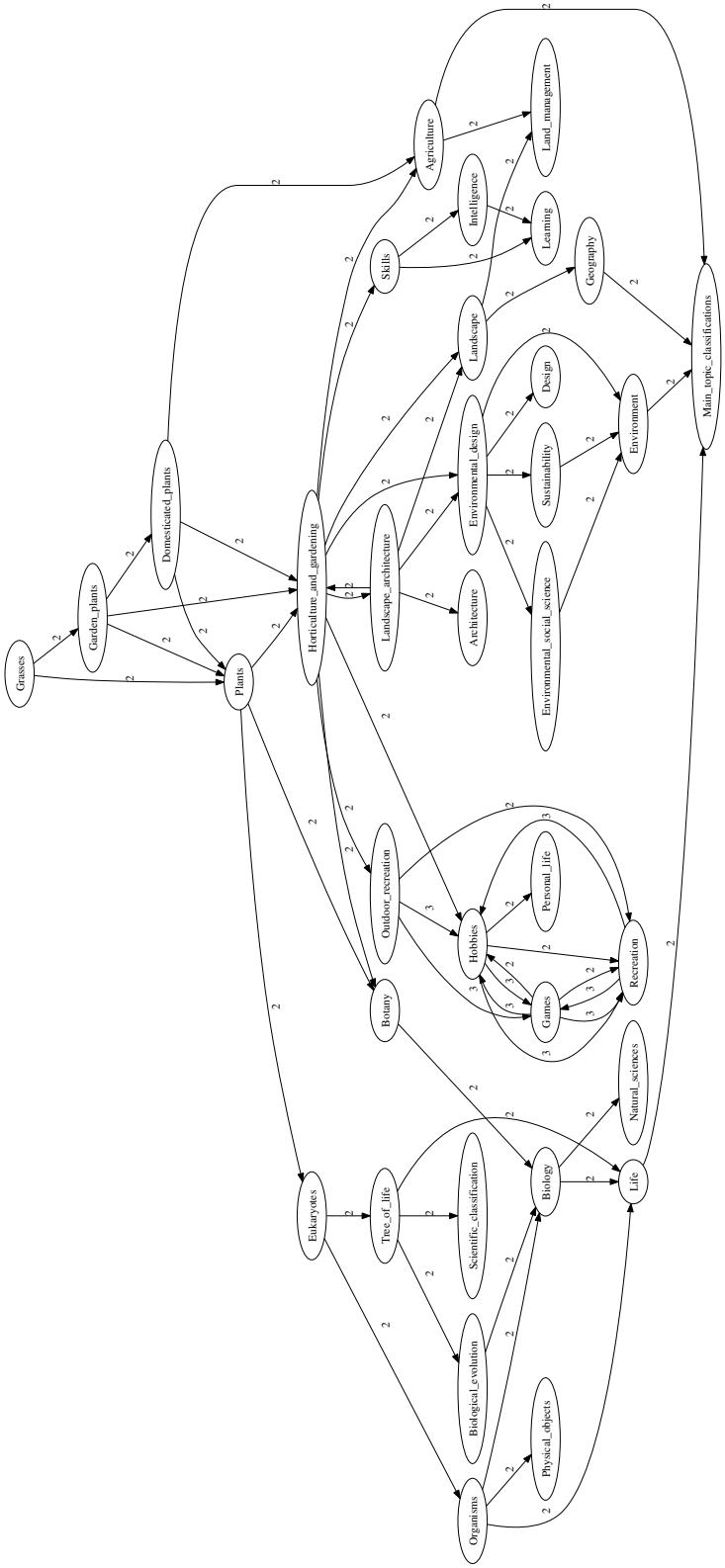

Figura 5.3: Muestra de relaciones entre categorías - *Grasses*

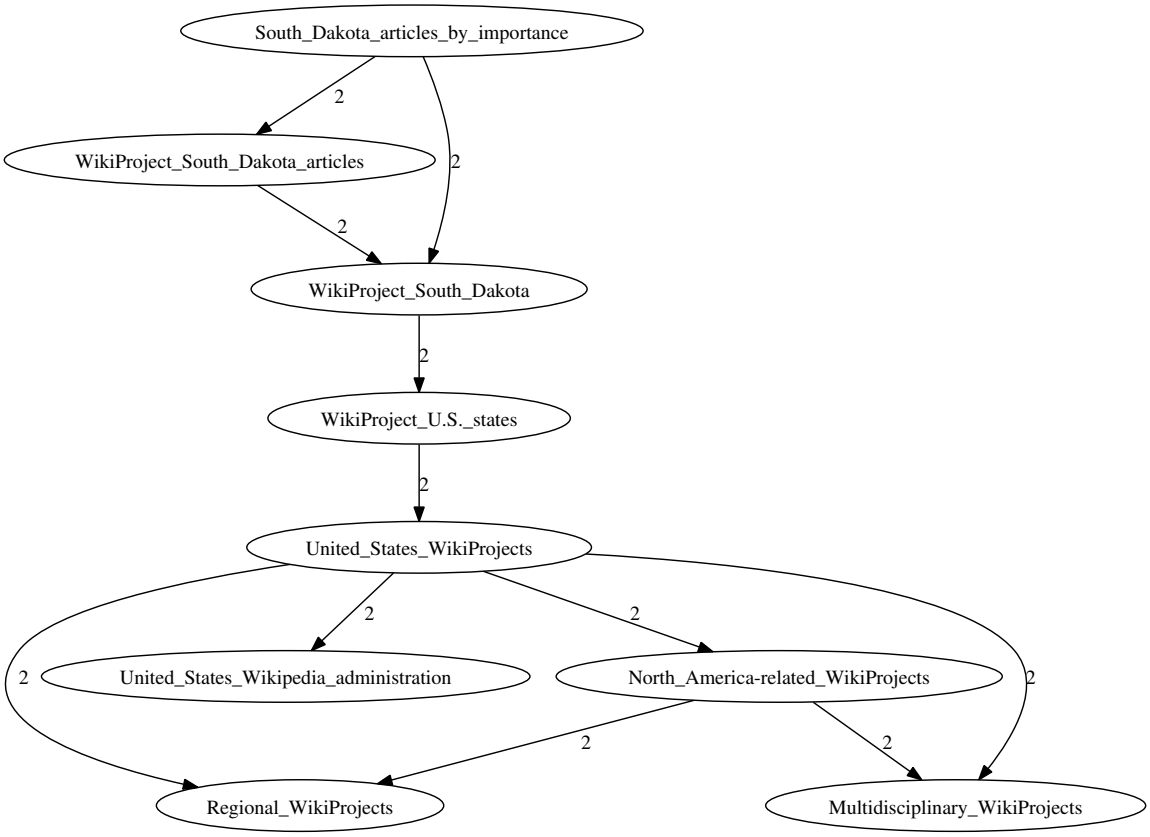

Figura 5.4: Muestra de relaciones entre categorías - *South Dakota articles by importance*.

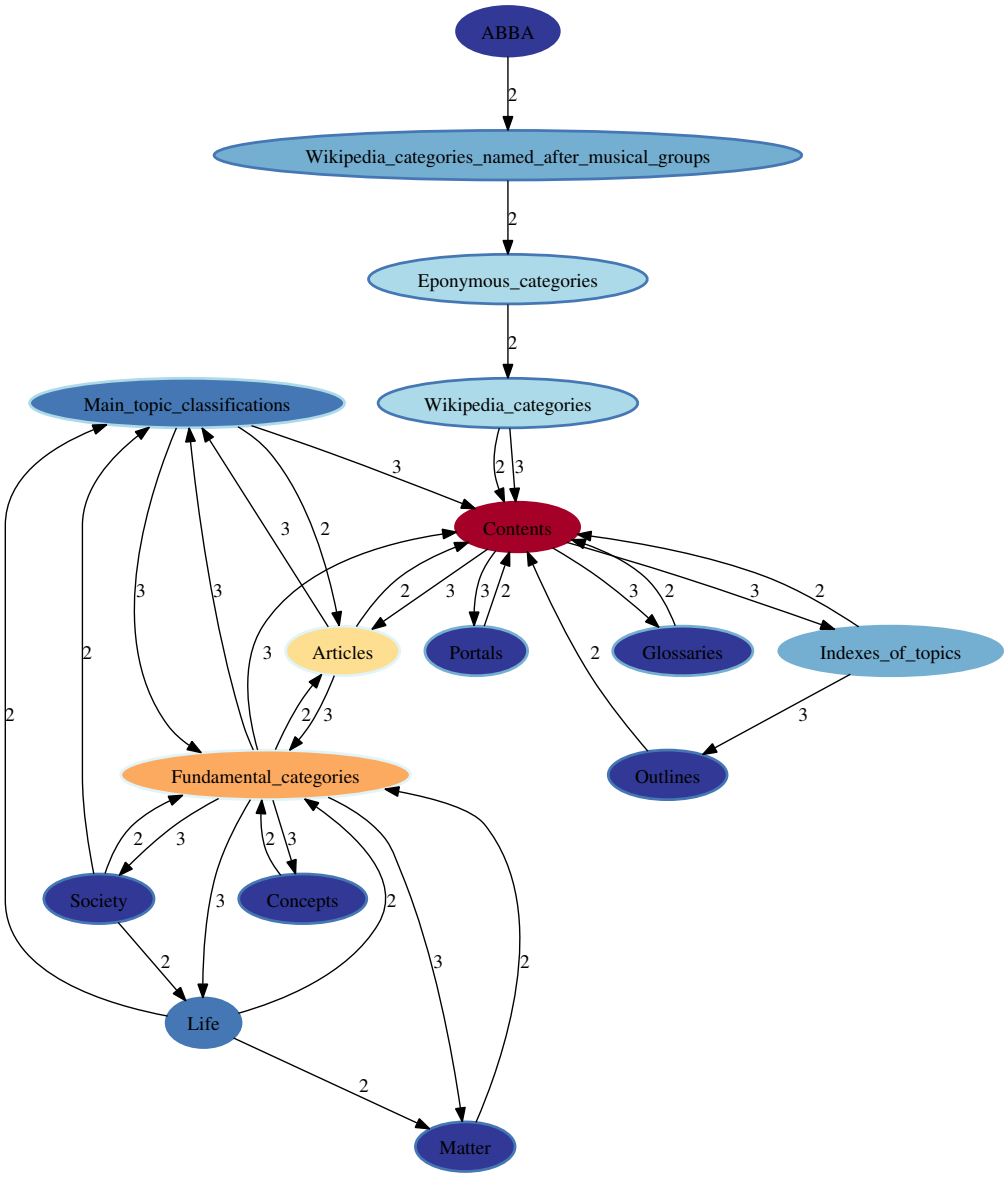

Figura 5.5: Muestra de relaciones entre categorías con centralidad y PageRank - *ABBA*

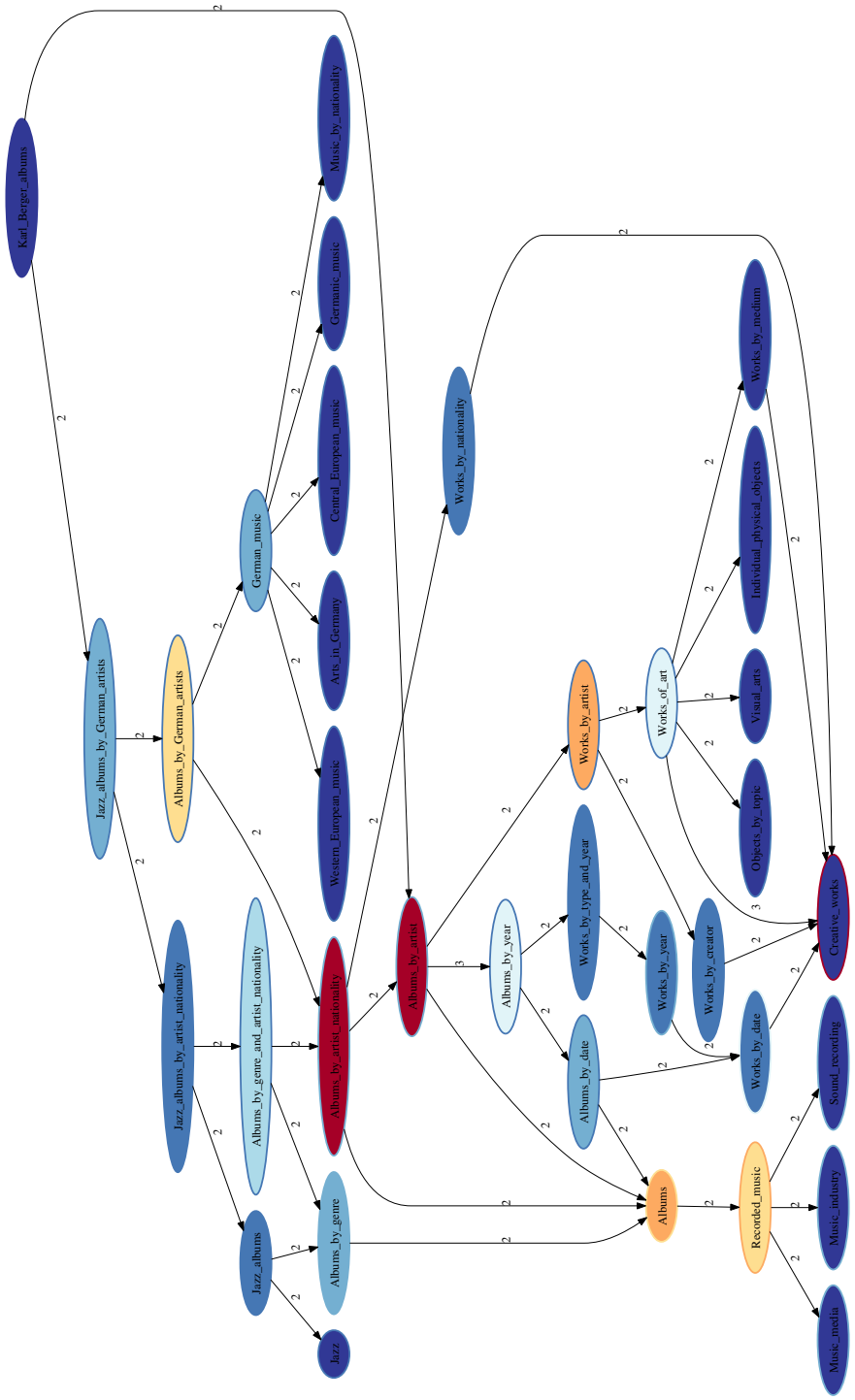

Figura 5.6: Muestra de relaciones entre categorías con centralidad y PageRank - *Karl Berger albums*

Capítulo 6

Validación de la Solución

Este capítulo se describe como se realizó la validación de la solución presentada en el Capítulo 4. Primero, se presenta el conjunto utilizado para estas pruebas. Luego, se explican los dos métodos que se utilizaron para obtención de muestras del conjunto de datos para realizar la experimentación. Posteriormente, se detallan los tres métodos de validación. Se detallan sus pasos y objetivos. Se finaliza el capítulo detallando los pasos seguidos en la experimentación y los resultados obtenidos junto al análisis de ellos.

6.1. Conjunto de datos - DBpedia

En el Capítulo 5 se detallaron los distintos conjuntos de datos que provee DBpedia. Se nombraron cuales son las ventajas y desventajas de cada uno. Finalmente, se decidió cual sería el conjunto de datos utilizado para el desarrollo de la solución, aquél que describe las relaciones entre las categorías de Wikipedia. Para la validación se utilizó el mismo conjunto de datos, ya que se deseaba observar el desempeño y calidad de la solución en un conjunto del mismo tipo de estructura.

6.2. Obtención de muestras

En la experimentación se necesitaban distintos conjuntos de datos donde aplicar la solución para formular una conclusión sobre ella. Por ello, se obtuvieron muestras del conjunto de datos para tener una experimentación con una cantidad considerable de experimentos, realizar una experimentación válida y así obtener resultados útiles para el análisis de la solución.

Para que una muestra del conjunto de datos fuera considerada buena, ésta debía ser aquella que representara al conjunto de la mejor manera, es decir, que su estructura fuera parecida y que los análisis que se le realizaran fueran característicos al conjunto. Por ello se determinaron dos métodos para la obtención de muestras: *BFS* y *Forest Fire*. Estos métodos

fueron escogidos para la obtención de muestras basándose en el trabajo de J. Leskovec y C. Faloutsos [14]. Con estos métodos se puede obtener muestras que mantienen de buena nivel las características del conjunto de datos y así poder hacer un estimado para el conjunto completo. Ambos métodos se detallan a continuación.

6.2.1. BFS

Este método para la obtención de muestras es búsqueda por amplitud (*Breadth-first search* en inglés). Procede de la siguiente manera: Se toma un nodo elegido uniformente al azar dentro de todo el conjunto de datos y aplicamos búsqueda en amplitud seleccionando nodos hasta obtener la cantidad de nodos deseados. Luego, la muestra es el subgrafo inducido a partir de los nodos seleccionados y el conjunto de datos. Un subgrafo H de un grafo G es inducido si para cada par de nodos v y w pertenecientes a H, vw es un arco del grafo H si y sólo si vw es un arco del grafo G. En otras palabras H es un grafo inducido de G si H tiene los mismos arcos que G tiene para el mismo conjunto de nodos.

6.2.2. Forest Fire

Este método para la obtención de muestras se basa en una idea bastante particular. La idea de este método es encender fuego en un nodo y esperar que se propague hasta que se hayan quemado suficientes nodos. Se elige al azar un nodo *semilla*, se le prende fuego y se comienza a propagar el fuego a través de los enlaces, "quemando" nodos vecinos. Si un nodo se quema, sus arcos y nodos vecinos tienen una chance de quemarse, aplicándose recursivamente. Luego, la muestra es el grafo inducido por los nodos quemados.

Entonces, para elegir un grafo de muestra con este método se procede de la siguiente manera: Se escoge un nodo uniformemente al azar. Generamos un número aleatorio x que es geométricamente distribuido con esperanza $\frac{7}{3}$. Se utiliza ese valor como esperanza ya que era el que mejor resultados presentó en el trabajo de J. Leskovec y C. Faloutsos [14]. Luego, se eligen x vecinos del nodo *semilla* uniformemente al azar. Se aplica el método recursivamente en los nodos elegidos hasta que se hayan quemado una cantidad suficiente de nodos. Si el "fuego" se apaga antes de completar la cantidad de nodos deseada, se elige otra *semilla* y se inicia el "fuego" nuevamente.

6.3. Métodos de validación

Los métodos para validar la solución propuesta son tres: Palabras claves, frecuencia de palabras y caminos aleatorios. Fueron elegidos para determinar la calidad de los resultados obtenidos en tres ámbitos. Con el método de palabras claves se espera obtener, a través de la evaluación de usuarios, en que medida se refieren los resultados de la solución a los temas generales de los datos. Con el método de frecuencia de palabras se espera determinar la relación entre el resultado de la solución y los conceptos más repetidos dentro del conjunto de

datos. Finalmente, con el método de caminos aleatorios se espera determinar la distribución de la solución respecto al conjunto de datos.

6.3.1. Palabras claves

Este método se basa en entregarle los resultados de la solución a usuarios para que evalúen la calidad de los resultados. Se diseñó para muestras pequeñas, entiéndase como pequeñas a muestras de no más de 200 nodos. Al basarse en evaluación por usuarios, muestras de mayor tamaño dificultan su análisis. Si las muestras tuvieran más de 200 nodos, el usuario tendría que lidiar con muchos datos al realizar la evaluación. Por ello, se definió una cantidad máxima de 200 nodos para que el usuario pudiera realizar la evaluación en un tiempo prudente, no más de media hora.

El método se divide en tres partes. Primero, una fase preliminar donde se recolectan las etiquetas correspondientes a los nodos que pertenecen a la muestra donde se aplicará la solución y se listan en orden lexicográfico. Segundo, se le entrega la lista ordenada a un usuario para que pueda elegir qué palabras claves considera importantes o relevantes dentro del listado con un máximo de 5. Tercer paso, el usuario toma los resultados de la solución propuesta y los compara. Ve qué tan relacionadas están las palabras claves que seleccionó con los resultados de la solución.

6.3.2. Frecuencia de palabras

Este método se diseñó para muestras sin importar su tamaño. Se divide en tres partes. Primero, se toman todas las etiquetas de los nodos que pertenecen a la muestra. Ya que se tienen las etiquetas, se separan las etiquetas en palabras, se eliminan las *stopwords*, se calcula la frecuencia de aparición de cada palabra y se ordenan de mayor a menor según la frecuencia recién calculada. Luego, tomamos las primeras palabras hasta completar una frecuencia menor o igual a 0,25. En la segunda parte, se eligen n nodos de la muestra de forma uniformemente aleatoria, siendo n la cantidad de nodos entregados por la solución. Finalmente, para la extracción aleatoria y para el resultado de la solución se calcula que porcentaje de las etiquetas contiene al menos una de las palabras escogidas.

6.3.3. Caminos aleatorios

Este método también se diseñó para muestras sin importar su tamaño. La idea es determinar qué tan 'cerca' está un nodo cualquiera de los entregados por la solución. Para ello procedemos de la siguiente manera: Dada la muestra, obtenemos el resultado de la solución. Luego, se elige un nodo uniformemente al azar de la muestra y, a partir de este nodo, realizamos una caminata aleatoria hasta llegar a un nodo que pertenezca a la solución. La caminata aleatoria la realizamos con una probabilidad de avanzar de $p_f = 0{,}8$ y de retroceder $p_b = 0{,}2$. Cuando se detiene la caminata, guardamos el largo del camino recorrido. Este resultado se

comparará con otras mediciones del grafo seleccionado como muestra, tales como el diámetro, densidad, camino más corto promedio y coeficiente de clustering.

6.4. Experimentación y resultados

Una vez determinado el conjunto de datos, la forma de obtener muestras y los métodos de validación, se procedió a realizar la experimentación. Se describirá la forma en que se hizo la experimentación con cada método, para luego mostrarse el análisis de los resultados, respondiendo las preguntas formuladas cuando se determinó cada método de validación.

Para la experimentación las muestras que se obtuvieron se dividieron en dos grupos: Muestras grandes y muestras pequeñas. El conjunto de muestras pequeñas se compone de 25 muestras elegidas con el método *BFS* y 25 con el método *Forest Fire*. En este grupo sólo se encuentran muestras de tamaño 200. Este fue el grupo utilizado para la primera experimentación, el método de palabras claves. Por otro lado, el grupo de muestras grandes se compone de 50 muestras obtenidas con el método *BFS* y 50 obtenidas con el método *Forest Fire*. Las muestras de este grupo se componen de muestras de tamaño que van desde 2000 a 35000 nodos. Este grupo se utilizará para la segunda y tercera experimentación, los métodos de frecuencia de palabras y caminos aleatorios respectivamente.

A continuación se presentan los procedimientos y resultados de cada experimentación.

6.4.1. Palabras claves

La validación con este método fue realizada por 10 usuarios. Ellos procedieron a obtener palabras claves de las muestras y compararlas con los resultados de la solución. En esta experimentación se utilizó el conjunto de muestras pequeñas. Se asignaron 5 muestras del grupo a cada usuario. De este modo las 50 muestras del grupo fueron evaluadas. Además, la asignación se realizó de forma aleatoria para que a un usuario le fuera asignada una muestra elegida con el método *BFS* o con el método *Forest Fire*.

Los usuarios eligieron en promedio 4,4 palabras claves de las etiquetas del grafo. Compararon las palabras claves con los resultados de la solución propuesta y llegaron a la siguiente conclusión. Las etiquetas de los nodos que entrega la solución propuesta representan en forma directa a las palabras claves en un 52,3 % en promedio. Y, el resto de las etiquetas que no representaron a las palabras claves en forma directa, sí lo hicieron en forma indirecta. Todas de las etiquetas de los nodos entregados por la solución representaron en forma directa o indirecta las palabras claves de los usuarios.

Entiéndase por 'aplicar directamente' a que la palabra clave es representada de una forma general, es decir, la etiqueta representa al tema de la misma manera que lo hace una palabra clave. Y, entiéndase por 'aplicar indirectamente' a que el tema que toca la etiqueta pertenece al que toca la palabra clave, pero de una forma más específica. Por ejemplo, consideremos que se tiene la palabra clave 'transporte', la etiqueta 'transporte por continente' y la etiqueta

'compañías de transporte obsoletas del Reino Unido'. Entonces, un usuario podría considerar que la palabra clave 'transporte' es representada en forma directa por la etiqueta 'transporte por continente', no así 'compañías de transporte obsoletas del Reino Unido' que representa a la palabra clave de forma indirecta.

6.4.2. Frecuencia de palabras

En esta experimentación se utilizó el grupo de muestras grandes. La ejecución de este método se dividió en 2 casos: Las muestras obtenidas con el método *BFS* y las obtenidas con *Forest Fire*. Para cada muestra se realizó el cálculo del método para el resultado de la solución y de la extracción aleatoria. Para el cálculo de la elección aleatoria, se repitió 30 veces, calculando el promedio. Luego de obtener los resultados para cada muestra se promediaron los valores para tener un valor por caso.

A continuación se describen los valores para cada caso y método de extracción.

Muestra	BFS	Forest Fire
Aleatoria	44.36 %	31.02 %
Solución	41.75 %	37.53 %

Tabla 6.1: Resultados método frecuencia de palabras.

6.4.3. Caminos aleatorios

Para esta experimentación se utilizó el conjunto de muestras grandes, dividiendo los casos en las muestras obtenidas por los métodos *BFS* y *Forest Fire*. Para cada una de las muestras de este conjunto se ejecutó el método de caminos aleatorios 100 veces y se calculó el promedio. De esta manera se obtuvo el largo promedio del camino aleatorio para cada muestra ($lpca$). Ya habiendo obtenido estos valores, se procedió a comprarlos con el diámetro (di$a$) y el valor del camino más corto promedio (ccp) de la muestra correspondiente. Se realizó la comparación a través de razones. Se obtuvo para cada muestra la razón entre $lpca$ y di$a$ y la razón entre $lpca$ y ccp. Además, se realizó la misma prueba cambiando los resultados de la solución por nodos elegidos uniformemente al azar dentro de la muestra. Se obtuvieron los siguientes valores:

Razón Muestra	$lpca$: di$a$	$lpca$: ccp
BFS - Solución	0.1144	0.2137
BFS - Aleatorio	0.6704	1.2775
Forest Fire - Solución	0.1106	0.2882
Forest Fire - Aleatorio	0.3141	0.8192

Tabla 6.2: Resultados método caminos aleatorios.

Los resultados en detalle se pueden observar en las secciones A.1, A.2, A.3 y A.4 en los anexos.

6.4.4. Análisis de Resultados

A partir de los resultados obtenidos de la experimentación se pueden concluir tres puntos importantes sobre los resúmenes entregados por la solución propuesta:

1. Son resúmenes que representan en un alto porcentaje a los temas generales que tocan los datos.
2. Son resúmenes que representan los temas populares sin sobrecargarse de ellos.
3. Son resúmenes bien distribuidos, no dejan temas aislados.

El primer punto se concluyó directamente de los resultados obtenidos. Es un buen resultado que los usuarios hayan considerado que, en promedio, un 52,3 % del resultado de la solución represente de forma directa a lo que ellos consideraron como palabras claves de los datos. Además, ellos mismos expresaron que el resto de los resultados de la solución corresponden a las palabras claves, aunque lo hicieron con temas más específicos. Con este punto podemos decir que se cumple el propósito de determinar de forma global los recursos más importantes.

Para el segundo punto, si observamos los resultados podemos ver que en el caso de muestras elegidas con *BFS* tenemos un 41,75 % de concordancia de las etiquetas con las palabras más frecuentes y para el caso de muestras elegidas con *Forest Fire* un 37,53 %. Estos son buenos resultados, ya que la solución no abarca sólo los temas que más se repiten, sino que también a temas menos populares. Si observamos los resultados de las extracciones aleatorias en las muestras elegidas con el método *Forest Fire*, podemos ver que éstas no superan en concordancia a la solución. Esto indica que la solución busca de una forma más específica qué nodos retornar.

Finalmente, podemos ver que los resultados de la solución entrega datos que no están concentrados en un sólo tema en específico o en un sector de los datos, sino que están bien distribuidos. La razón entre el largo promedio del camino aleatorio ($lpca$) y el diámetro de la muestra es de 0,1144 en las muestras elegidas con *BFS* y de 0,1106 en las muestras elegidas con *Forest Fire*. Por otro lado, si observamos la razón con el camino más corto promedio, vemos que las razones aumentan. Tenemos las razones 0,2137 y 0,2882 para las muestras elegidas con *BFS* y *Forest Fire* respectivamente. Estas razones muestran que la distancia entre un recurso cualquiera y uno de entregado por la solución es pequeña, lo cual hace que tenga más probabilidad de ser representado por el recurso de la solución.

Uniendo estos tres puntos tenemos que la solución entrega resultados que hace un balance entre entregar recursos generales, populares y con una buena distribución dentro de los datos.

Capítulo 7

Conclusiones y Trabajo Futuro

En este capítulo se presentan las conclusiones de esta memoria. Se presenta el trabajo realizado y los objetivos alcanzados. Además, se expone el trabajo futuro, explicando que mejoras se pueden hacer al trabajo realizado o qué puntos vale la pena profundizar en esta investigación.

7.1. Conclusiones

En ese trabajo de memoria se diseñó e implementó un método para la realización de resúmenes semiautomáticos de un conjunto de datos RDF. Dado un grafo RDF se puede obtener un conjunto de nodos, cuyo tamaño es determinado por el usuario, el cual representa y da a entender cuales son los temas más importantes dentro del conjunto completo. Este método fue diseñado en base a los conjuntos de datos provistos por DBpedia.

Se logró establecer que los resultados entregados por el método presentado tuvieran un buen balance de los temas generales, temas populares y la distribución respecto al conjunto de datos completo.

Se logró mostrar que la centralidad de intermediación junto a los grados son suficientes para determinar recursos importantes de una forma simple. Esto es muy relevante a la hora de escalar este método semiautomático a conjuntos de datos más grandes.

Los resultados obtenidos a través de los tres métodos de experimentación son más que satisfactorios e indican la calidad del método, habiendo realizado una evaluación a nivel de usuario y evaluación automática.

Se lograron los objetivos generales y específicos establecidos el inicio de este trabajo. Se cumplió con el objetivo general de analizar distintas soluciones para realizar resúmenes semiautomáticos y la implementación de ésta. Se cumplió, también con los objetivos específicos de este trabajo. Se analizaron nociones de relevancia en trabajos relacionados. Se determinó el conjunto de datos donde se trabajaría. Se diseñó e implementó una solución a partir

de las ideas preliminares. Y, por último, se realizó experimentación de la solución para su evaluación.

7.2. Trabajo futuro

A pesar de haberse cumplido los objetivos propuestos con el trabajo, quedan muchas cosas por mejorar y/o explorar que surgieron en el curso de esta investigación.

Primero, probar el método en otros tipos de conjuntos de datos que formalicen conocimiento y ver qué resultados produce. Ver si aplica de la misma manera que con los datos de DBpedia y cuáles son sus diferencias. Segundo, utilizar y comparar resultados con otros algoritmos de centralidad, distintos a la centralidad de intermediación, por ejemplo, centralidad de cercanía o centralidad Eigenvector. La centralidad de intermediación es la que mejor se ajustaba a los datos de DBpedia, pero en otros conjuntos de datos puede que no sea así.

Y, finalmente, investigar más sobre algoritmos de arbolización para poder aplicar este procedimiento en el método diseñado.

Finalmente, uno de los temas que quedó abierto con los resultados de esta memoria, es la utilidad de los métodos de arbolización, que en teoría funcionan muy bien, pero hasta hoy en día parecen ser inalcanzables en la práctica debido a su costo computacional.

Bibliografía

[1] DBpedia. `http://www.dbpedia.org`.

[2] Resource Data Framework. `http://www.w3c.org/RDF`.

[3] World Wide Web Consortium. `http://www.w3c.org`.

[4] K. Anyanwu, A. Maduko, and A. Sheth. Semrank: Ranking complex relationship search results on the semantic web. In *Proceedings of the 14th international conference on World Wide Web*, pages 117–127. ACM, 2005.

[5] Grady Booch. *Object oriented design with applications*. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1991.

[6] U. Brandes. A faster algorithm for betweenness centrality*. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[7] S.N. Chander, V.S. Kumar, and P.R. Prasath. Enhancing the relevance of semantic web information retrieval results using extension theory. In *Trendz in Information Sciences & Computing (TISC), 2010*, pages 218–221. IEEE, 2010.

[8] L. Ding, R. Pan, T. Finin, A. Joshi, Y. Peng, and P. Kolari. Finding and ranking knowledge on the semantic web. *The Semantic Web–ISWC 2005*, pages 156–170, 2005.

[9] Robert W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345, 1962.

[10] L.C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[11] Vibhav Gogate and Rina Dechter. A complete anytime algorithm for treewidth. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, UAI '04, pages 201–208, Arlington, Virginia, United States, 2004. AUAI Press.

[12] A. Graves, S. Adali, and J. Hendler. A method to rank nodes in an RDF graph. In *The 7th International Semantic Web Conference, Oct*, pages 26–30, 2008.

[13] Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 538–543, New York, NY, USA, 2002. ACM.

[14] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636. ACM, 2006.

[15] T. van Dijk, J.P. van den Heuvel, and W. Slob. Computing treewidth with libtw, 2006.

Apéndice A

Resultados Caminos Aleatorios

En las siguientes tablas se muestran los resultados de la experimentación con el método de caminos aleatorios. Abreviaciones:

- **nnodes**: Cantidad de nodos en la muestra.
- **s-nodes**: Cantidad de nodos en el resultado de la solución.
- **a-nodes**: Cantidad de nodos seleccionados al azar.
- **stdDev**: Deviación estándar de los resultados de los caminos aleatorios para una muestra.
- **average**: Promedio de los resultados de los caminos aleatorios para una muestra.
- **dia**: Diámetro de la muestra.
- **apl**: Promedio de los caminos más cortos de la muestra.
- **cc**: Coeficiente de clustering de la muestra.
- **den**: Densidad de la muestra.

A.1. Muestras elegidas con método BFS y resultados solución

#	nnodes	s-nodes	stdDev	average	dia	apl	cc	den
1	1590	318	0.9215	0.9700	6	2.4967	0.004054	0.002175
2	16316	3264	0.8292	1.1500	8	5.2245	0.003763	0.000224
3	1538	308	0.7871	1.0200	6	2.5190	0.004097	0.002213
4	18436	3688	0.9035	1.0600	10	5.4931	0.008183	0.000242
5	29864	5973	1.0050	1.1000	12	5.9790	0.012798	0.000148
6	3174	635	0.8476	0.9600	10	3.9021	0.003908	0.001029
7	2152	431	0.6621	1.0400	6	2.8607	0.004106	0.001693
8	15069	3014	0.3544	0.8800	11	4.5502	0.000316	0.000221
9	1457	292	0.6756	0.9400	8	3.5275	0.007706	0.002578
10	13348	2670	0.5362	0.8500	8	3.7412	0.000085	0.000267
11	3314	663	0.8515	0.9300	8	4.2492	0.005301	0.001046
12	6702	1341	1.3865	1.2400	8	5.3328	0.022961	0.000662
13	30641	6129	1.0569	1.2300	10	5.9244	0.004980	0.000126
14	14652	2931	0.8888	1.0100	10	6.2447	0.025254	0.000249
15	24928	4986	0.8136	1.0900	12	6.0316	0.005570	0.000150
16	20246	4050	0.7348	1.0000	10	5.4123	0.001298	0.000192
17	31833	6367	0.4477	0.8600	8	4.7325	0.000086	0.000106
18	14676	2936	0.8999	1.0100	8	4.9073	0.007727	0.000273
19	1583	317	0.7613	0.9800	6	3.0110	0.009871	0.002419
20	21310	4262	1.0660	1.0600	9	5.2348	0.004292	0.000180
21	26110	5222	0.7906	1.0700	8	4.9347	0.003468	0.000196
22	17981	3597	0.8209	1.3100	10	5.4835	0.004078	0.000204
23	7191	1439	0.5173	0.8200	8	3.8583	0.006427	0.000565
24	7836	1568	0.8874	1.1500	10	5.7243	0.017552	0.000509
25	21439	4288	0.7723	0.9400	11	6.3449	0.001021	0.000156
26	13547	2710	1.1839	1.2800	10	5.4979	0.004747	0.000263
27	30381	6077	1.0945	1.1100	12	5.9937	0.006277	0.000146
28	30513	6103	0.7732	1.1100	10	5.4579	0.002525	0.000146
29	30244	6049	0.7274	1.0300	10	5.5054	0.006202	0.000143
30	21197	4240	0.9514	1.0700	11	6.3370	0.005672	0.000182
31	30883	6177	0.4308	0.8800	8	4.4157	0.000911	0.000124
32	34144	6829	0.4176	0.8400	8	4.0889	0.000042	0.000116
33	13191	2639	0.4771	0.8200	10	5.6813	0.004802	0.000231
34	33845	6769	0.6581	0.8700	10	5.9952	0.002298	0.000104
35	33012	6603	0.8364	1.0200	10	5.8298	0.007388	0.000135
36	10688	2138	0.5895	0.9500	8	4.3620	0.005082	0.000379
37	21634	4327	0.4555	0.8500	10	4.9960	0.001440	0.000191
38	2970	594	1.1720	1.0800	6	3.3951	0.006064	0.001396
39	11881	2377	0.4556	0.8200	12	5.8392	0.006695	0.000257
40	1884	377	0.7219	0.8300	6	3.1892	0.004451	0.001853
41	18984	3797	0.6161	0.9800	10	5.5182	0.002330	0.000179
42	2811	563	0.6633	1.0000	8	3.5290	0.004054	0.001244
43	9459	1892	0.5674	0.9100	8	4.5864	0.003728	0.000433
44	30535	6107	0.8367	1.0000	10	5.4416	0.004215	0.000156
45	4696	940	0.4668	0.8900	8	3.9309	0.004978	0.000710
46	5603	1121	0.7060	0.9600	8	4.5116	0.004511	0.000668
47	26216	5244	0.4770	0.8500	10	5.7282	0.005116	0.000134
48	6882	1377	0.6756	0.9400	8	3.7521	0.006446	0.000595
49	8033	1607	0.4711	0.9100	8	4.0631	0.006632	0.000512
50	33900	6780	1.2411	1.1400	10	5.5850	0.008517	0.000136

Tabla A.1: Muestras elegidas con método BFS y resultados solución

A.2. Muestras elegidas con BFS y resultados aleatorios

#	n-nodes	a-nodes	stdDev	average	dia	apl	cc	den
1	1590	318	7.7695	3.7100	6	2.4967	0.004054	0.002175
2	16316	3264	8.2392	6.4200	8	5.2245	0.003763	0.000224
3	1538	308	9.1667	6.4700	6	2.5190	0.004097	0.002213
4	18436	3688	5.7948	4.9800	10	5.4931	0.008183	0.000242
5	29864	5973	6.5930	5.5500	12	5.9790	0.012798	0.000148
6	3174	635	10.6233	7.1600	10	3.9021	0.003908	0.001029
7	2152	431	9.9524	7.5200	6	2.8607	0.004106	0.001693
8	15069	3014	5.9771	5.7100	11	4.5502	0.000316	0.000221
9	1457	292	5.2377	4.1300	8	3.5275	0.007706	0.002578
10	13348	2670	8.0106	6.4800	8	3.7412	0.000085	0.000267
11	3314	663	9.3871	6.7700	8	4.2492	0.005301	0.001046
12	6702	1341	8.0087	5.9800	8	5.3328	0.022961	0.000662
13	30641	6129	7.7098	6.1400	10	5.9244	0.004980	0.000126
14	14652	2931	9.2758	6.3300	10	6.2447	0.025254	0.000249
15	24928	4986	7.5894	6.0200	12	6.0316	0.005570	0.000150
16	20246	4050	8.2223	7.1200	10	5.4123	0.001298	0.000192
17	31833	6367	9.2818	7.7800	8	4.7325	0.000086	0.000106
18	14676	2936	9.4820	6.5400	8	4.9073	0.007727	0.000273
19	1583	317	4.6002	3.2800	6	3.0110	0.009871	0.002419
20	21310	4262	6.8374	5.3600	9	5.2348	0.004292	0.000180
21	26110	5222	5.8229	4.7100	8	4.9347	0.003468	0.000196
22	17981	3597	7.3552	5.9800	10	5.4835	0.004078	0.000204
23	7191	1439	7.4182	6.1000	8	3.8583	0.006427	0.000565
24	7836	1568	6.7552	5.1300	10	5.7243	0.017552	0.000509
25	21439	4288	8.2631	8.0400	11	6.3449	0.001021	0.000156
26	13547	2710	6.7056	4.4300	10	5.4979	0.004747	0.000263
27	30381	6077	7.0696	6.0200	12	5.9937	0.006277	0.000146
28	30513	6103	7.3068	5.5000	10	5.4579	0.002525	0.000146
29	30244	6049	7.2816	5.4100	10	5.5054	0.006202	0.000143
30	21197	4240	7.7401	5.9900	11	6.3370	0.005672	0.000182
31	30883	6177	6.6697	5.6600	8	4.4157	0.000911	0.000124
32	34144	6829	5.9935	5.2400	8	4.0889	0.000042	0.000116
33	13191	2639	9.5109	7.1100	10	5.6813	0.004802	0.000231
34	33845	6769	6.3701	5.3200	10	5.9952	0.002298	0.000104
35	33012	6603	7.2574	6.4800	10	5.8298	0.007388	0.000135
36	10688	2138	7.0177	4.8500	8	4.3620	0.005082	0.000379
37	21634	4327	7.5752	5.4200	10	4.9960	0.001440	0.000191
38	2970	594	8.0372	7.2700	6	3.3951	0.006064	0.001396
39	11881	2377	9.4132	6.6500	12	5.8392	0.006695	0.000257
40	1884	377	6.9743	5.6700	6	3.1892	0.004451	0.001853
41	18984	3797	6.3196	5.3200	10	5.5182	0.002330	0.000179
42	2811	563	9.0021	7.7700	8	3.5290	0.004054	0.001244
43	9459	1892	5.5857	4.0000	8	4.5864	0.003728	0.000433
44	30535	6107	7.1626	6.2400	10	5.4416	0.004215	0.000156
45	4696	940	7.2876	6.9700	8	3.9309	0.004978	0.000710
46	5603	1121	7.2002	5.2400	8	4.5116	0.004511	0.000668
47	26216	5244	7.9418	5.7800	10	5.7282	0.005116	0.000134
48	6882	1377	6.5423	4.6700	8	3.7521	0.006446	0.000595
49	8033	1607	4.3357	4.0400	8	4.0631	0.006632	0.000512
50	33900	6780	5.9216	5.5700	10	5.5850	0.008517	0.000136

Tabla A.2: Muestras elegidas con BFS y resultados aleatorios

A.3. Muestras elegidas con método Forest Fire y resultados solución

#	stdDev	average	dia	apl	cc	den
1	1.9936	1.8400	19	6.4792	0.035403	0.000214
2	2.2227	1.8600	19	7.1507	0.028856	0.000215
3	2.2640	2.1200	16	6.7422	0.054854	0.000215
4	2.2069	1.6400	18	7.4514	0.013498	0.000207
5	3.3087	2.3500	18	6.7487	0.027561	0.000220
6	2.7363	1.9500	16	7.0025	0.034429	0.000213
7	1.8539	1.7300	19	6.8485	0.019821	0.000201
8	2.9143	2.3700	18	6.8649	0.028096	0.000205
9	4.2778	2.2000	21	8.0170	0.032616	0.000193
10	2.6393	2.2100	17	7.0511	0.022002	0.000191
11	2.9687	1.8700	19	6.3670	0.024816	0.000204
12	3.4742	2.3600	21	8.2561	0.025962	0.000191
13	2.9097	2.5600	18	6.8965	0.055513	0.000212
14	2.7556	2.3700	18	6.9154	0.032281	0.000202
15	3.1447	2.4700	17	6.8281	0.030716	0.000194
16	2.3731	1.7800	18	6.7914	0.005797	0.000214
17	2.6363	1.9900	17	6.5157	0.036339	0.000223
18	2.0385	1.6200	19	7.4054	0.018482	0.000207
19	1.7256	1.6800	16	5.8379	0.020382	0.000224
20	2.1831	1.7100	18	6.7999	0.028745	0.000209
21	3.0256	2.1600	20	7.8705	0.015266	0.000196
22	1.8830	1.8800	18	7.1012	0.022293	0.000221
23	2.6687	2.0900	19	7.2120	0.005203	0.000213
24	2.7425	2.3300	20	7.9450	0.018659	0.000200
25	1.9507	1.9300	18	6.6414	0.026668	0.000204
26	1.4384	1.4700	17	6.8074	0.004746	0.000221
27	1.9631	1.9200	19	6.9382	0.041991	0.000213
28	2.3879	1.9100	17	6.4533	0.027543	0.000208
29	3.5162	2.5800	18	7.7364	0.037511	0.000191
30	1.6447	1.5700	18	6.4857	0.019616	0.000196
31	3.1381	2.1500	18	7.1397	0.053585	0.000207
32	1.9556	1.6600	22	8.5900	0.026945	0.000194
33	2.3579	1.9800	19	7.0176	0.038056	0.000205
34	1.7397	1.4400	17	6.4618	0.019683	0.000196
35	1.6062	1.6000	16	6.2237	0.027059	0.000213
36	2.2045	1.6000	17	7.0617	0.069055	0.000213
37	3.0431	2.1400	19	7.1036	0.008479	0.000214
38	1.8466	1.7000	17	6.8136	0.032062	0.000213
39	2.8961	2.1800	18	6.8740	0.038417	0.000201
40	4.7958	3.0000	20	7.2585	0.040689	0.000198
41	3.2538	2.1500	19	6.8334	0.023862	0.000195
42	2.6758	2.2000	18	7.2427	0.029867	0.000194
43	2.3954	1.8900	20	7.4094	0.026378	0.000190
44	2.2180	2.0200	20	7.3747	0.011224	0.000204
45	2.6933	1.6900	15	5.7422	0.018639	0.000216
46	2.6268	2.4000	20	7.4967	0.033871	0.000191
47	2.4921	2.3600	18	6.4412	0.028122	0.000208
48	3.2721	2.4400	22	8.4965	0.026212	0.000189
49	2.4310	1.7000	20	7.3059	0.022337	0.000196
50	3.1909	2.4100	17	6.8498	0.030273	0.000200

Tabla A.3: Muestras elegidas con método Forest Fire y resultados solución

A.4. Muestras elegidas con método Forest Fire y resultados aleatorios

#	stdDev	average	dia	apl	cc	den
1	7.6451	5.6500	19	6.4792	0.035403	0.000214
2	7.9070	5.8000	19	7.1507	0.028856	0.000215
3	8.7229	5.5300	16	6.7422	0.054854	0.000215
4	7.3048	6.0200	18	7.4514	0.013498	0.000207
5	7.1824	5.6500	18	6.7487	0.027561	0.000220
6	7.1642	6.4300	16	7.0025	0.034429	0.000213
7	5.4290	5.1600	19	6.8485	0.019821	0.000201
8	9.8196	6.3400	18	6.8649	0.028096	0.000205
9	7.9612	6.6000	21	8.0170	0.032616	0.000193
10	6.8844	4.6200	17	7.0511	0.022002	0.000191
11	9.1296	6.3600	19	6.3670	0.024816	0.000204
12	7.0974	6.6300	21	8.2561	0.025962	0.000191
13	7.1711	5.0700	18	6.8965	0.055513	0.000212
14	9.0216	6.5100	18	6.9154	0.032281	0.000202
15	7.6438	5.8500	17	6.8281	0.030716	0.000194
16	8.7053	6.7600	18	6.7914	0.005797	0.000214
17	11.9367	8.2900	17	6.5157	0.036339	0.000223
18	7.0969	5.7100	19	7.4054	0.018482	0.000207
19	5.9370	5.5500	16	5.8379	0.020382	0.000224
20	8.6054	5.3700	18	6.7999	0.028745	0.000209
21	10.1906	6.5400	20	7.8705	0.015266	0.000196
22	6.4536	4.9700	18	7.1012	0.022293	0.000221
23	6.2408	4.8200	19	7.2120	0.005203	0.000213
24	5.9653	5.3400	20	7.9450	0.018659	0.000200
25	6.8640	5.3100	18	6.6414	0.026668	0.000204
26	7.2375	5.9100	17	6.8074	0.004746	0.000221
27	6.8062	5.6600	19	6.9382	0.041991	0.000213
28	7.4391	5.4000	17	6.4533	0.027543	0.000208
29	7.5787	5.9400	18	7.7364	0.037511	0.000191
30	7.2499	5.3300	18	6.4857	0.019616	0.000196
31	4.6842	4.2800	18	7.1397	0.053585	0.000207
32	6.7040	5.5800	22	8.5900	0.026945	0.000194
33	7.6285	5.8100	19	7.0176	0.038056	0.000205
34	6.8386	5.5600	17	6.4618	0.019683	0.000196
35	6.2658	5.2000	16	6.2237	0.027059	0.000213
36	7.1376	5.2100	17	7.0617	0.069055	0.000213
37	6.6353	5.6500	19	7.1036	0.008479	0.000214
38	8.2448	6.2300	17	6.8136	0.032062	0.000213
39	8.1266	6.0900	18	6.8740	0.038417	0.000201
40	6.3076	5.7100	20	7.2585	0.040689	0.000198
41	7.6600	5.6200	19	6.8334	0.023862	0.000195
42	6.5067	5.3200	18	7.2427	0.029867	0.000194
43	5.8232	4.9700	20	7.4094	0.026378	0.000190
44	5.7325	4.3300	20	7.3747	0.011224	0.000204
45	6.8557	5.8300	15	5.7422	0.018639	0.000216
46	6.8081	5.1000	20	7.4967	0.033871	0.000191
47	6.9781	6.3100	18	6.4412	0.028122	0.000208
48	8.3713	6.3900	22	8.4965	0.026212	0.000189
49	7.4928	5.9100	20	7.3059	0.022337	0.000196
50	7.5030	6.1600	17	6.8498	0.030273	0.000200

Tabla A.4: Muestras elegidas con método Forest Fire y resultados aleatorios