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

AUTOMATIZACIÓN DE REFERENCIAS EN WIKIDATA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

PAOLO LUCCA CUROTTO MOLINA

PROFESOR GUÍA:
AIDAN HOGAN

MIEMBROS DE LA COMISIÓN:
BÁRBARA POBLETE LABRA
JOSÉ URZÚA REINOSO

SANTIAGO DE CHILE
2020

RESUMEN

Desde su creación Wikidata se enfocó en dar solución a los problemas que enfrenta Wikipedia sobre la estructuración de la información que guarda. Ha experimentado un rápido crecimiento desde su incepción en 2012, transformándose en una fuente de información valiosa e interconectada de la web que es usada por varias aplicaciones. Todos los días se agregan elementos y hechos nuevos que guardan información sobre el mundo real, sin embargo, aún enfrenta un desafío para el cual no se tiene una solución definitiva: la adición de fuentes para la procedencia de la información. Al ser una fuente secundaria de información requiere que todo lo afirmado en ella esté debidamente respaldado, indicando la fuente de donde se obtuvo dicha información. Pero actualmente no cuentan con ninguna herramienta automática que realice este trabajo, por lo que la búsqueda de referencias queda en manos de los colaboradores que las busquen manualmente para las afirmaciones que agreguen. El no contar con un método fácil para agregar referencias ha causado que el 34 % de las afirmaciones no están respaldadas. Esto presenta un problema ya que cuando afirmaciones no cuentan con información sobre su procedencia, se corre el riesgo de no saber si esa información es correcta ya que no se puede validar. Por consiguiente los datos pierden credibilidad y valor para quien quiera hacer uso de ellos.

Para intentar dar solución a este problema se desarrolló una herramienta que entrega referencias a hechos de Wikidata de forma automática. Esta permite buscar elementos de Wikidata y luego encontrar posibles referencias para cualquier hecho que se elija. Para cada afirmación se retornan 3 URLs de páginas autoritativas donde aparece información del hecho junto con un pequeño extracto de la página en donde aparecen los términos de la afirmación. Para esto se utilizó un índice de documentos construido usando las páginas de la sección de Referencias de los artículos de Wikipedia. Para encontrar referencias se consulta este índice usando el método TF-IDF en donde los términos de búsqueda son: el nombre del elemento, de la propiedad y del valor del hecho al que se está buscando referencias.

Se probó el rendimiento usando elementos con distintas cantidades de afirmaciones y referencias en Wikipedia, y se encontró en promedio que solo el 37 % de las afirmaciones lograban ser validadas por alguna referencia de Wikipedia. Sin embargo para aquellas que sí tenían una referencia que la validaba el método TF-IDF logró ser efectivo en encontrar aquellos documentos relevantes para la afirmación. Demostrando ser un buen paso para ayudar a encontrar referencias pero aún queda trabajo por hacer para lograr encontrar una solución definitiva.

Tabla de Contenido

Capítulo 1

Introducción

1.1. Contexto

Una de las páginas más populares de la web es Wikipedia, la enciclopedia online, que contiene más de 40 millones de artículos en 301 idiomas diferentes. Esto es una cantidad tremenda de información estructurada en base a documentos de texto. Organizarla para hacerla mantenible ha sido un desafío desde su concepción, ya que al tener tantas versiones para un mismo artículo, se tiene la misma información repetida en distintos idiomas. Uno de los problemas que esto presenta es que la información debe ser constantemente actualizada para todos los idiomas, sin generar discrepancias entre versiones diferentes. Se dieron cuenta que en vez de tener la misma información en 200 versiones para un mismo artículo, era necesario tenerla centralizada en un solo lugar.

Por esto, en el año 2012 nació Wikidata [19], la plataforma que tiene Wikipedia para estructurar su información. En un principio, por cada artículo de Wikipedia se creaba un elemento en Wikidata. Este elemento servía para guardar información básica como el nombre, descripción y enlaces en los distintos idiomas en los que estaba disponible ese artículo. Posteriormente se añadieron afirmaciones a los elementos de Wikidata. Estas corresponden a pares propiedad-valor que permiten definir a los elementos de forma estructurada, a diferencia de su versión en Wikipedia que es en base a texto. Por ejemplo, el elemento "Universidad de Chile" tiene definida para la propiedad "fecha de fundación" el valor "1842", y para la propiedad "rector" el valor "Ennio Vivaldi", el cual a su vez es un elemento de Wikidata.

Wikidata se transformó un repositorio de información de Wikipedia. Mucha de la información que se guardaba en forma de texto y se repetía para los distintos idiomas de Wikipedia, se trasladó a Wikidata para ser importada desde allí. Como los sitelinks o enlaces a otros idiomas de los artículos, que debían copiarse para cada idioma en los que un artículo estaba disponible, actualmente esa información está centralizada en Wikidata. Wikipedia también importa información que es usada en los infoboxes de los artículos; esta información es fácilmente reutilizable entre las distintas versiones, por lo que Wikidata sirve también como repositorio para guardar este tipo datos.

Pero Wikidata no solo guarda información de artículos de Wikipedia. Con el trabajo de miles de colaboradores y datos donados de otras bases de datos como Freebase [2], Wikidata se transformó en una gran fuente de información interconectada que define al mundo real y sus propiedades. Es un grafo de conocimiento colaborativo, mantenido por sus usuarios en su totalidad, y actualmente su comunidad cuenta con 21.000 colaboradores activos mensuales. Desde su creación, se han añadido más de 70.000.000 elementos y 890.000.000 afirmaciones. Además de proveer información a Wikipedia, dentro de todos los proyectos de la Wikimedia Foundation, Wikidata es usado en un 57 % de las páginas, siendo Wikipedia la que en cantidad más utiliza datos, mostrando información obtenida de Wikidata en 67 % de las páginas[1]. También es usada en servicios externos: Wikidata es uno de los proveedores de datos al Knowledge Graph de Google [17, 16], y es parte de aplicaciones de voz que responden a preguntas de personas como Siri de Apple [11], entre otras.

Wikidata es una fuente abierta para cualquier persona o aplicación que quiera hacer uso de sus datos. Ha adoptado los principios de Linked Data [7] y actualmente cuenta con un endpoint SPARQL público [6] y está disponible también en versión RDF [5].

1.2. Problema

Wikidata, al igual que Wikipedia, es una fuente secundaria de información, es decir, todo lo afirmado allí es solo una recolección desde otras fuentes. Por esta razón, toda afirmación en Wikidata debe ser verificable por una referencia a una fuente que respalde dicha afirmación. Esta referencia debe cumplir con ser *autoritativa*, es decir, debe ser considerada confiable, estar actualizada y estar libre de sesgo. Esta se agrega como un par propiedad-valor a nivel de afirmación, en donde el valor corresponde a la fuente.

Cuando una afirmación cuenta con un referencia pasa a llamarse un hecho. En general todas las afirmaciones deben estar respaldadas; solo existen algunas excepciones que no necesitan referencia como hechos indisputables o cuando una afirmación solo conecta al elemento a una base de datos externa mediante un ID. Para todas las demás afirmaciones, los colaboradores deben agregar al menos una referencia, interna o externa, que la respalde. Si una afirmación no cuenta con referencia, o la referencia no la valida correctamente, Wikidata se da el derecho a eliminar esta afirmación. A pesar de esto, existen aproximadamente 306 millones que no tienen referencia válida, lo que corresponde a un 34 % del total de 890 millones de referencias que existen en Wikidata.

[1]Porcentaje de artículos que hacen uso de los datos de Wikidata `http://wdcm.wmflabs.org/WD_percentUsageDashboard/`

1.3. Situación Actual

Si bien la labor más importante de Wikidata es representar la información de Wikipedia en elementos de forma estructurada, como fuente secundaria de información, es indispensable la buena calidad de la procedencia de la información. Sin referencias confiables, la calidad de los datos inmediatamente disminuye su valor, especialmente en Wikidata, que es un servicio de libre edición y hay aplicaciones que dependen de sus datos.

Actualmente no existe ningún programa que añada referencias automáticamente, por lo que los colaboradores deben buscarlas manualmente en las páginas de Wikipedia o Google. Lo más cercano a una herramienta automática para ayudar a los colaboradores es la "Primary Sources Tool"[2]. Esta herramienta sugiere de forma automática afirmaciones y referencias a hechos que obtiene de una base de datos externa importada como Freebase [12]. El problema es que esta herramienta depende de que colaboradores importen bases de datos externas para que ésta pueda sugerir referencias a nuevos hechos.

También cuentan con un buscador de fuentes de libros[3]. Ingresando un ISBN[4], la página provee enlaces a catálogos de bibliotecas, bases de datos y otros sitios con más información acerca del libro. Sin embargo, es difícil pensar que esta solución sea efectiva, ya que no otorga una manera fácil de buscar referencias a algún hecho. A pesar de que los libros son en general una buena fuente de referencias, son muy imprácticos para esa tarea, debido a su largo y en algunos casos a su complejidad. En la práctica, para un colaborador es mucho más fácil buscar referencias en páginas web.

[2]Página de Wikidata del Primary Sources Tool: `https://www.wikidata.org/wiki/Wikidata:Primary_sources_tool`

[3]El buscador de fuentes de libros está disponible en `https://www.wikidata.org/wiki/Special:BookSources`

[4]El *International Standard Book Number* es un identificador numérico de libros

1.4. Objetivos

1.4.1. Objetivo General

Crear una herramienta que permita ayudar a los colaboradores de Wikidata a encontrar referencias autoritativas a hechos de forma automática, sugiriendo, para cualquier afirmación, una lista de posibles referencias que puedan validarla, y así ayudar a disminuir la cantidad de hechos que no tienen referencia en Wikidata.

1.4.2. Objetivos Específicos

1. Crear una aplicación web que permita buscar elementos de Wikidata según su identificador, entregando la lista de todos los hechos asociados a ese elemento.
2. Dado algún hecho, entregar una lista de referencias junto con un breve extracto en donde se valida el hecho al cual estamos buscando referencias.
3. Validar el desempeño de las sugerencias. Es necesario evaluar si las referencias que sugiere el programa son en general correctas y aceptables como fuentes para los hechos.

1.5. Descripción General de la Solución

La solución escogida corresponde a una automatización de la primera estrategia que propone Wikidata para encontrar referencias: buscar en la sección de referencias de las páginas de Wikipedia de los respectivos elementos de Wikidata. Para la automatización del proceso, la solución se basa en un índice que contiene todas las páginas que aparecen en la sección de Referencias de todos los artículos de la versión en inglés de Wikipedia, como se muestra en la figura 1.1. Este índice se construye usando Apache Nutch™[5] para descargar y parsear documentos, y Apache Solr™[6] para indexarlos. Luego para buscar referencias a algún hecho de algún elemento de Wikidata, se consulta este índice utilizando el algoritmo de ranking por defecto de Solr, que se basa en el método TF-IDF [15]. Para los términos de la consulta se usan el nombre del elemento, el nombre del hecho y el valor del hecho, obteniendo como resultado las 3 mejores páginas, ordenadas por relevancia, en donde se pueda encontrar una posible referencia que respalde ese hecho. La figura 1.2 muestra un esquema de un ejemplo de este proceso.

Para cada documento o referencia que se guarda en este índice se almacenan, entre otras cosas, el contenido de la página web y los identificadores de los elementos de Wikidata de las respectivas páginas de Wikipedia en donde aparece esa referencia. Por ejemplo, si una página aparece en la sección de referencias de los artículos de Wikipedia de Chile y Santiago, entonces para esa página se guardan los valores [298, 2887], ya que en Wikidata Chile corresponde al identificador "Q298" y Santiago a "Q2887". La búsqueda de los términos mediante TF-IDF se realiza sobre el campo del documento que tiene el contenido de las páginas. Los identificadores se guardan para que la búsqueda de documentos también se pueda acotar a las referencias de la página de Wikipedia de cada elemento. Hacer esto puede mejorar los resultados, especialmente cuando se buscan conceptos muy generales en el índice. Evidentemente, para acotar la búsqueda es necesario que el elemento de Wikidata tenga contraparte en Wikipedia, la cual además debe tener una cantidad razonable de enlaces en su sección de referencias. Otra opción para mejorar los resultados consiste en extender los términos de consulta. Agregando los nombres alternativos de los elementos, propiedades y valores que provee Wikidata, se aumentan las alternativas para encontrar buenos resultados. Sin embargo no es una medida que se pueda usar en todos los casos, ya que tiene el costo de que se agreguen posiblemente referencias irrelevantes a los resultados.

La idea era que la solución funcionase de forma *online*, es decir, para cada elemento al que se le buscaba referencias se visitaba su respectiva página de Wikipedia, se extraían las referencias, se descargaban los documentos y se añadían al índice. Este proceso puede demorar varios minutos, y va a depender en mayor medida de la cantidad de referencias que se tenga que descargar, ya que es la parte más lenta de realizar. Esperar mucho tiempo por cada elemento al que se le quiera buscar referencias no es lo ideal, por lo que para minimizar la cantidad de documentos que se debían descargar se contaban con dos medidas:

1. Antes de descargar las páginas obtenidas de Wikipedia, se verifican aquellas que ya se

[5] *Well matured, production ready, highly extensible & scalable Web crawler* `https://nutch.apache.org/`
[6] *Popular, blazing-fast, open source, enterprise Search platform* `https://lucene.apache.org/solr/`

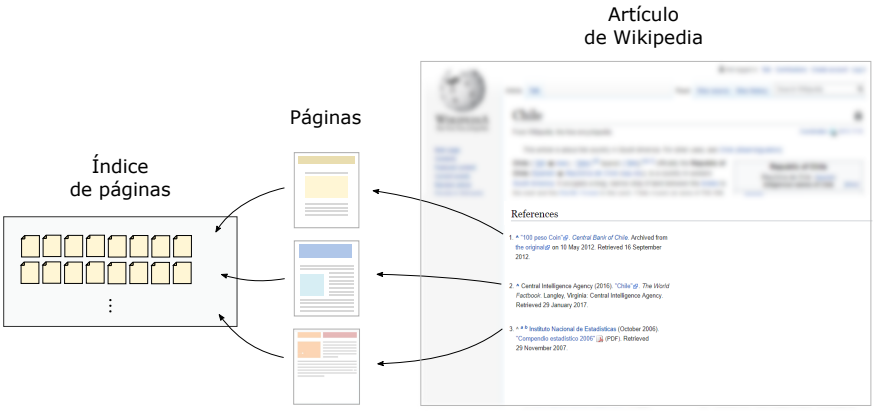

Figura 1.1: La solución se basa en una índice páginas autoritativas. Estas se obtienen de las secciones de referencias de los artículos de Wikipedia.

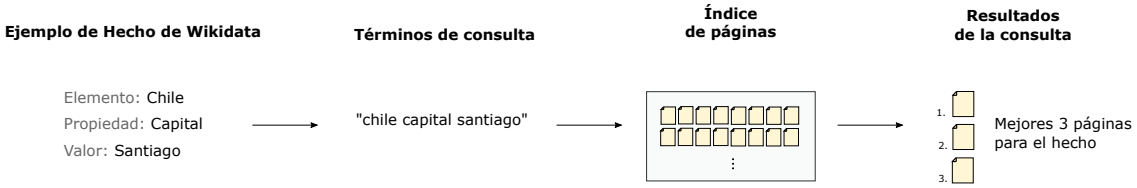

Figura 1.2: Esquema del proceso para buscar referencias a un hecho de Wikidata. A partir de la información del hecho, se generan los términos de búsqueda y se consulta el índice usando el algoritmo de Solr, obteniendo como resultado las páginas más relevantes para el hecho.

encuentran en el índice para no descargarlas nuevamente.

2. Añadir al índice con antelación la mayor cantidad de páginas como sea posible.

Estas dos medidas pretendían ayudar a disminuir el tiempo que tarda el proceso de obtener las referencias de Wikipedia para añadirlas al índice. Sin embargo, aún así, el tiempo a esperar es demasiado como para que sea parte de una solución usable. Por esto se decidió que el sistema fuera *offline*, es decir, que no involucrará descargar páginas. Simplemente se buscan referencias en los documentos que ya se encuentran en el índice, saltándose el proceso de visitar la página de Wikipedia, extraer referencias y descargar los documentos. Con esto un usuario no tendría que esperar nada para buscar referencias. Qué tan bien funcione el sistema usando este método va a depender si el índice contiene documentos relevantes al elemento que se está buscando referencias, porque si no contiene ninguno, los resultados serán muy malos. La idea es que el usuario obtenga resultados inmediatos y la herramienta sea usable, aunque esto afecte el rendimiento de la solución.

La herramienta podrá quedar disponible para que sea usada por cualquier colaborador de Wikidata. En un principio se pensó que fuera añadida directamente a Wikidata como *plugin*, en caso de ser aceptada por Wikidata, ya que por ejemplo va en contra del principio de Wikidata de ser plurilingüe al retornar solo referencias en inglés para las afirmaciones. Pero también podría funcionar como sistema independiente en caso de demostrar ser útil y contar con la infraestructura necesaria.

Capítulo 2

Marco Teórico

La implementación de este trabajo se basa en el uso de web crawling y tecnologías de la web semántica como Linked Data. En esta sección primero se hablará del concepto de Linked Data, junto con sus tecnologías RDF y SPARQL: un modelo de datos para representar información y lenguaje de consultas para acceder a datos RDF respectivamente. También se dará una explicación de Wikidata sobre cómo funciona, su relación con Wikipedia y cómo ha adoptado los principios de Linked Data. Finalmente se comentarán los principios de web crawling para extraer información de la web y sus limitaciones.

2.1. Linked Data

La forma tradicional de guardar y enlazar información en la web es mediante documentos HMTL. Este formato permite navegar la web mediante enlaces y poder publicar información de forma fácil. Pero a su vez esto tiene sus desventajas al momento de querer buscar o acceder a información, ya que ésta se almacena en formato de texto, y por lo tanto es difícil poder extraer automáticamente información específica y relevante. En general las páginas definen información sobre distintos conceptos de forma desordenada en el documento, y los algoritmos de indexación necesitan leerlo todo para recién saber de qué trata la página.

El concepto de Linked Data se refiere a una manera distinta de guardar información en la web, esto es de forma interconectada y estructurada. Los datos que pertenecen al Linked Data de la web cumplen con:

- Ser leíbles por máquinas.
- Tener definición explícita.
- Estar conectados a otras fuentes de datos.
- Poder ser enlazados desde otras fuentes de datos.

Para lograr lo anterior se necesita algo distinto a lo convencional que son documentos HTML basados en texto. Es por esto que Linked Data define datos usando el Resource Description Framework (RDF).

2.1.1. RDF

Una de las características de RDF es que permite definir información de forma estructurada. A diferencia de HTML, cada archivo define de forma separada entidades o conceptos del mundo real. Las entidades corresponden a recursos identificados mediante URIs que pueden ser accedidas mediante el protocolo HTTP.

La forma en que se definen las cosas en RDF es usando tripletas sujeto-predicado-objeto. El sujeto es siempre una URI que identifica un recurso, el predicado es también una URI que especifica cómo se relaciona el sujeto con el objeto. El objeto puede ser una URI o un literal.

Por ejemplo, imaginemos que queremos representar la siguiente información:

```
- La FCFM es una facultad de la U. de Chile

- La U. de Chile y la U. Católica son universidades

- Las universidades son instituciones de educación superior
```

En formato RDF/Turtle se puede escribir usando 4 tripletas:

```
@prefix   rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns\#> .
@prefix   rdfs:  <http://www.w3.org/2000/01/rdf-schema\#> .
@prefix   ej:    <http://www.ejemplo.cl/\> .

# Definición de tripletas
ej:FCFM ej:facultad_de ej:universidad_de_chile .
ej:universidad_de_chile rdf:type ej:universidad .
ej:universidad_catolica rdf:type ej:universidad .
ej:universidad rdfs:subClassOf ej:institucion_de_educacion_superior .
```

La definición de estas tripletas equivalen a un grafo, el cual se puede visualizar en la figura 2.1.

Para definir las relaciones se usaron predicados que ofrecen los vocabularios RDF y RDFS (RDF Schema) [3]. Para definir que un objeto *es* alguna otra cosa se usa generalmente el predicado “rdf:type” y para decir que algo pertenece a otra cosa se utiliza “rdfs:subClassOf”. En este caso “rdf:” y “rdfs:” corresponden a prefijos. En general es recomendable usar predicados de vocabularios existentes y no crear relaciones nuevas cuando estas ya están definidas,

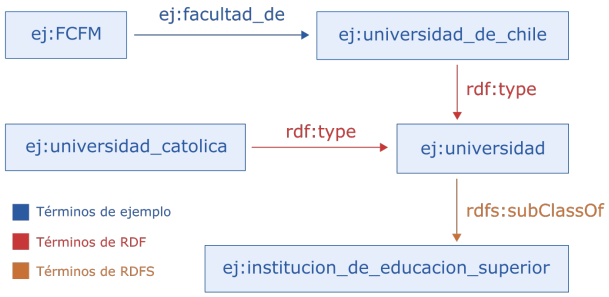

Figura 2.1: Ejemplo de grafo RDF

ya que sino estaríamos duplicando información. Existen muchos vocabularios utilizados por usuarios del Linked Data, como OWL (Web Ontology Language) [8] que fue inventado con el propósito de crear relaciones para describir clases y sus relaciones; o el vocabulario FOAF (Friend of a Friend) [4], que es usado para describir relaciones entre personas, entre otros.

En este caso los objetos como "FCFM" y "Universidad de Chile" se definieron como recursos pertenecientes a un dominio imaginario solo a modo de ejemplo, usando el prefijo "ej". En la práctica, uno define sus recursos en su propio dominio. Uno también puede crear sus propias relaciones; para representar que la FCFM es una facultad de la Universidad de Chile se definió la relación "facultad de". En este ejemplo se usaron solo objetos del mismo dominio, pero en realidad uno es libre de conectar sus objetos con otros de distintos sets de datos.

La idea fundamental del Linked Data es que los datos definidos por medio de RDF se pueden ver como un grafo de recursos esparcidos por la web donde cualquier persona puede añadir más información. Todos pueden acceder y recorrer este grafo, y al encontrar datos desconocidos, se pueden desreferenciar las URI para para encontrar la definición de estos.

Al publicar datos se deben seguir las siguientes reglas [1]:

1. Usar URI's para nombrar las cosas.
2. Usar URI's HTTP para que otras personas puedan encontrar esos nombres.
3. Cuando busque alguna URI, se debe proveer información relevante.
4. Incluir enlaces a otras URI's RDF, para que se pueda navegar el grafo y descubrir más cosas.

Estos se conocen como los principios de Linked Data, cuya función es hacer que la información publicada pertenezca a un espacio global común y facilitar el acceso por parte de programas o personas a los datos de la web.

2.1.2. SPARQL

Para facilitar el acceso a los datos del Linked Data se creó SPARQL, es el lenguaje para consultar y obtener datos de bases de datos cuya información esté en formato RDF. SPARQL permite hacer consultas en formato objeto-propiedad-valor retornando resultados que coinciden con el patrón de datos del grafo RDF que se está consultando. Por ejemplo, para obtener las instituciones de educación superior de los datos del ejemplo de la sección 2.1.1, se debe ejecutar la siguiente consulta (formato RDF/Turtle):

```
PREFIX   rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
PREFIX   ej:  <http://www.ejemplo.cl/> .

SELECT ?institucion
WHERE {
     ?institucion  rdf:type ej:institucion_de_educacion_superior .
}
```

La cual entrega estos resultados:

institución
<http://www.ejemplo.cl/universidad_de_chile>
<http://www.ejemplo.cl/universidad_catolica>

Este es solo un pequeño ejemplo, pero las capacidades de SPARQL son mucho mayores. Las consultas también permiten conjunciones, disyunciones, patrones opcionales, agregación, sub-consultas, negación, etc.

2.2. Grafo de Conocimiento

En el año 2012 Google anunció el Google Knowledge Graph, cuya función era mejorar los resultados de su motor de búsqueda, esto se ve reflejado en la caja de información que aparece al costado derecho de los resultados de una búsqueda. Por ejemplo, al buscar el nombre de una persona famosa, en esta caja se muestra información sobre su fecha de nacimiento, edad, profesión, entre otras cosas. Según Google, se trata de un modelo inteligente que entiende las entidades del mundo real y sus relaciones. El concepto fundamental era pasar de una búsqueda que se basaba solo en palabras a una que fuera a nivel de entidades concretas como personas, países, películas, bandas, en el sentido en que los humanos entendemos esas cosas. Con la introducción del Knowledge Graph, Google al mismo tiempo creó el término "Grafo de conocimiento". Un grafo de conocimiento describe entidades del mundo real y sus interrelaciones, organizando esta información en forma de grafo. La gracia de hacerlo de esta manera es que el grafo adquiere y junta datos en una ontología y puede deducir a partir de

ella más información. Puede ser usado en aplicaciones como búsqueda semántica, chatbots inteligentes y recomendaciones basadas en contenido, entre otras. Actualmente existen varios grafos de conocimientos disponibles en la web como DBpedia [10], YAGO2 [9] y Wikidata. Para generar estas bases de conocimiento y que sean útiles es necesario que contengan grandes volúmenes de datos. Los dos primeros grafos mencionados son creados a partir de información extraída de Wikipedia, mientras que Wikidata es editada por sus contribuidores. Dado que este trabajo se enfoca en Wikidata, se discutirá este grafo en más detalle.

2.3. Wikidata

Wikidata nació como una solución para manejar, estructurar y acceder a la información contenida en los artículos de Wikipedia. Actualmente es un grafo de conocimiento abierta que guarda información sobre entidades del mundo real y sus interrelaciones de forma estructurada. Comparte principios similares a los de Wikipedia en términos de ser de libre acceso, colaborativa, plurilingüe y fuente secundaria de información.

La primera versión de Wikidata fue lanzada en octubre del 2012. En ella la labor de los colaboradores fue crear elementos en Wikidata para cada artículo que existía en Wikipedia, para poder centralizar los enlaces de los distintos idiomas en los que estos estaban disponibles. Antes de que existiera Wikidata, estos enlaces se guardaban directamente en cada artículo de Wikipedia en formato de texto, lo cual generaba problemas de mantenibilidad y mucha repetición de texto. Además de almacenar estos enlaces, a los elementos se le podía agregar información básica como un nombre, etiquetas alternativas y una descripción de una sola línea.

Luego en el año 2013 se añadieron afirmaciones, estas son pares propiedad-valor que permiten definir y conectar a los elementos. En un principio el valor de las propiedades solo podían ser imágenes de Wikimedia Commons u otros elementos de Wikidata. Progresivamente se fueron añadiendo otros tipos de posibles valores para las propiedades como coordenadas, fechas y strings.

En el año 2015 se lanzó una de las herramientas más importantes de Wikidata en su misión de ser una fuente de información abierta, el Wikidata Query Service. Es un endpoint SPARQL que permite hacer consultas a los datos actualizados de Wikidata.

2.3.1. Modelo de Datos

La información en Wikidata se modela usando elementos y propiedades, las cuales se conocen como entidades. Los elementos se refieren a cualquier cosa del mundo real, sean objetos concretos o conceptos abstractos, como personas, ciudades, canciones, eventos históricos, etc. Mientras que las propiedades se utilizan para poder definir a los elementos, dándoles valores literales, o bien conectándolos con otros elementos. Las entidades en Wikidata tienen un

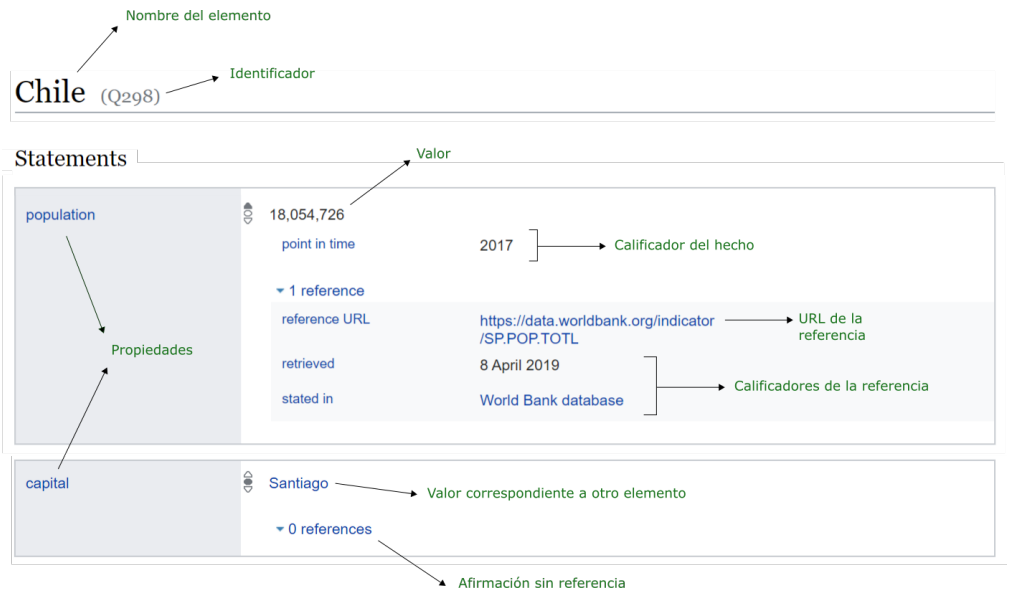

Figura 2.2: Nomenclatura de los distintos elementos presentes en una página de Wikidata

identificador único que está fijado para que sea independiente de cualquier idioma. El identificador de los elementos corresponde a una "Q" seguido por un número mientras que para las propiedades es una "P" seguido por un número.

Tanto elementos como propiedades tienen una URI o página única donde se puede ver su información, por ejemplo para acceder a la página del elemento "Chile", cuyo identificador es Q298, hay que acceder a:

`https://www.wikidata.org/wiki/Q298`

Mientras que para acceder a la información de la propiedad "Población", cuyo identificador es P1082, hay que acceder a:

`https://www.wikidata.org/wiki/Property:P1082`

En las páginas de todas entidades se puede encontrar la siguiente información:

- Datos básicos como nombre, descripción y etiquetas alternativas.
- Una lista de *sitelinks*.
- Una lista de afirmaciones.

Los datos básicos están principalmente para identificar y buscar elementos de forma fácil. Estos pueden existir en cualquier idioma disponible en Wikidata.

La lista de sitelinks corresponde a una lista de enlaces a los distintos proyectos de la Wikimedia Foundation y los idiomas en los que está disponible la entidad. Por ejemplo, en el caso de un elemento que corresponde a un artículo de Wikipedia, aquí se guardan los enlaces a los distintos idiomas en los que está disponible el artículo. Estos son los que aparecen en la barra lateral izquierda en las páginas de Wikipedia.

La lista de afirmaciones corresponde a la definición de los datos de la entidad. Una afirmación es un par propiedad-valor en donde la propiedad es una entidad de Wikidata y el valor puede tomar uno de estos tipos de datos:

- *Elemento*: Corresponde a la URI de un elemento de Wikidata.
- *Cantidad*: Para referirse a un valor numérico.
- *Tiempo*: Usado para fechas.
- *String*: Secuencia de letras y números en ningún idioma específico.
- *Identificador externo*: Identificador en algún sistema obase de datos externa.
- *Propiedad*: Corresponde a la URI de una propiedad de Wikidata.
- *Commons media*: Referencias a archivos de Wikimedia Commons.

Estos son los tipos de valores más comunes que pueden tomar las propiedades, pero también existen otros como: Texto monolingüe, Coordenada Terrestre, URL, Expresión matemática, Forma geográfica, Notación musical, Datos tabulares, Lexemas, Forma y Sentido. Las propiedades tienen restricciones sobre el tipo de valor que pueden tomar, por ejemplo, para la propiedad "Población" el valor debe ser una Cantidad y para "Fecha de creación" debe ser un Tiempo.

La figura 2.2 muestra un extracto de la página de "Chile" en donde se ven la mayoría de los elementos que conforman las afirmaciones. En ella aparecen dos, "Población" y "Capital", la primera que contiene una referencia y la segunda no. En la figura no se muestran las etiquetas alternativas ni la lista de sitelinks . Para el caso de la información sobre la población de Chile, primero se creó y añadió la siguiente afirmación al elemento "Chile" (Q298):

`Población (P1082)` $\Rightarrow$ `18.054.726 (Cantidad)`

Con esto se definió concretamente que Chile tiene una propiedad que es "Población" la cual a su vez sólo permite que su valor sea una "Cantidad", en este caso 18.054.726. Las afirmaciones no necesariamente deben tener un único valor, aún cuando se refieran a valores específicos, como en el caso de la población, ya que por ejemplo estas pueden referirse a distintos valores de la propiedad a través del tiempo. Al agregar varios valores para una propiedad, a veces es necesario especificar información adicional para precisar a qué se está refiriendo o bien para extender su definición. En el ejemplo anterior se menciona la población

de Chile, pero no se da información sobre la validez temporal de esa afirmación. En este caso la población es un valor que cambia en el tiempo, por lo que es recomendable precisar la fecha o año a la que ese valor se refiere. Para solucionar lo anterior existen calificadores, estos son pares propiedad-valor que se pueden agregar a las afirmaciones para darles información adicional a estas. En este caso se añadió un calificador para informar la fecha en la cual se hizo la medición de la población, usando la propiedad "Fecha". A la afirmación se le agregó el siguiente calificador para decir que el valor corresponde al año 2017:

`Fecha (P585)` $\Rightarrow$ `2017 (Tiempo)`

Una afirmación puede tener una cantidad indefinida de calificadores para extender su información. En el ejemplo anterior se podrían agregar otros como "Método de determinación" (P459) para precisar que se obtuvo el valor mediante estimación o censo, también "Población masculina" (P1540) y "Población femenina" (P1539) para separar los valores de la población por sexo, etc.

El segundo aspecto que permiten los calificadores es agregar referencias a las afirmaciones. Las referencias son un tipo especial de calificadores, ya que estas corresponden a 1 o más calificadores agrupados cuyo objetivo es ser una fuente de información de la afirmación. La propiedad a usar para definir la referencia va a depender si es externa o interna. Usar elementos existentes en Wikidata como fuentes de información corresponde a referencias internas, para las cuales se debe usar la propiedad "Afirmado en" (P248) y el valor debe ser el elemento de Wikidata en donde se valida la afirmación. Al utilizar referencia externas, se tiene que usar la propiedad "URL de la referencia" (P854) en donde el valor es, como el nombre lo indica, la dirección web de la fuente de información. Existen distintas reglas para agregar referencias dependiendo de si es interna o externa. En este caso se explicarán las referencias externas, ya que son las más comunes en Wikidata y más fáciles de agregar. Como se mencionó, una vez que se tiene una URL que se quiera usar como referencia, se debe agregar un calificador a la afirmación con la propiedad "URL de la referencia" (P854) y donde el valor es la URL. Además de eso, Wikidata sugiere que se agreguen los siguientes calificadores para rastrear la URL, en caso que ésta cambie:

- "Fecha de publicación" (P577): fecha de publicación de la página. En caso que esta información no esté disponible usar "Fecha de acceso" (P813) de cuando se accedió a la URL.
- "Título" (P1476): título de la página web.
- "URL de archivo" (P1065): dirección web de archivo de la misma página.
- "Fecha de archivo" (P2960): fecha de cuando se archivó el documento.
- "Idioma de la obra" (P407): idioma de la página web.

En caso de ser necesario, también se deben agregar estos calificadores a la referencia:

- "Autor" (P50) y "Editorial" (P123).
- "Cita" (P1683): extracto de texto exacto de la página.

Al agregar una referencia a una afirmación, ésta pasa a ser un trozo íntegro de información, la cual pasa a llamarse un hecho.

Siguiendo el ejemplo anterior de la población de Chile, para validar esta información se le agregó la siguiente referencia (grupo de calificadores):

```
URL de la referencia (P854) ⇒ https://data.worldbank.org/indicator/SP.POP.TOTL
Fecha de acceso (P813) ⇒ 8 de abril de 2019
Afirmado en (P248) ⇒ Base de datos del Banco Mundial (Q21540096)
```

Con lo cual la información sobre la población de "Chile" queda completa con su respectiva fuente (ver figura 2.2).

2.3.2. Wikidata en la Web Semántica

Wikidata es una base de información para los proyectos de la Wikimedia Foundation pero también para cualquiera que quiera hacer uso de ellos. En ese sentido, para hacer sus datos accesibles se ha basado en los principios de Linked Data. Wikidata cuenta con su información codificada en RDF, y provee dumps públicos para acceder a la base de datos completa en formatos adicionales como JSON y XML[1].

Además de esto, cuenta con el Servicio de Consultas de Wikidata, el cual se anunció en el año 2015. Permite a los usuarios ejecutar consultas sobre los datos contenidos en Wikidata a través de un endpoint SPARQL construido sobre la base de datos de grafos BlazeGraph [18]. Hay 2 formas de usar este servicio, programáticamente haciendo consultas GET o POST o bien mediante una interfaz web interactiva[2]. En ésta página se pueden escribir consultas en SPARQL, la cual entrega los últimos resultados disponibles en Wikidata. El editor está especialmente diseñado para funcionar con datos de Wikidata, proveyendo tooltips y compleción automática para identificadores de elementos y propiedades. Los resultados son mostrados de forma tabular, aunque también se han agregado varios plugins para extender la visualización de los datos consultados. El servicio permite visualizar datos mediante gráficos (líneas, barras, área, dispersión, dimensiones, burbujas), líneas de tiempo, grafos, árboles y mapas de árboles. También se pueden visualizar resultados que corresponden a puntos geográficos en mapa.

Además de este servicio de consultas, Wikidata ofrece una gran variedad de herramientas

[1]Descargas de la base de datos completa se puede obtener de `https://www.wikidata.org/wiki/Wikidata:Database_download`

[2]La interfaz web está disponible en `https://query.wikidata.org/`. Las consultas GET o POST se deben hacer a `https://query.wikidata.org/sparql`

para editar, consultar y visualizar datos[3]. Existen al menos 29 herramientas realizadas por colaboradores que permiten realizar consultas a los datos de distintas formas.

2.4. Web Crawling

Parte importante de la solución se basa en web crawling para bajar e indexar documentos que podrían servir como referencias, por lo que es importante mencionar algunos conceptos importantes al utilizarlo.

Web crawling se refiere al proceso de descargar páginas web en grandes cantidades. Se usa principalmente para crear motores de búsqueda, ya que al indexar los documentos permiten consultarlos y obtener resultados de forma rápida. Algunos como Google, Bing, Yandex, etc. realizan todos los días web crawling para mantener las páginas que han indexado actualizadas y también ir agregando aquellas nuevas que van apareciendo.

Todos los días la gente navega la web visitando unas cuantas páginas y descargando su contenido para visualizarlo; este proceso se hace sin pensarlo demasiado. Sin embargo, si alguien quisiera descargar páginas en gran escala, hay que pensar en el modo en que se va a realizar, ya que es deseable hacerlo de manera eficiente y es necesario seguir ciertas reglas. Los documentos en la web se pueden imaginar como un grafo conectado por enlaces; la idea es ir atravesando este grafo e ir descargando las páginas. Suponiendo una escala considerable de crawling, va a ser necesario paralelizar el proceso usando threads o haciéndolo de forma distribuida, ya que de esta manera se puede aumentar considerablemente la cantidad de páginas descargadas por unidad de tiempo, debido a que, en general, establecer las conexiones remotas toma mucho tiempo, por lo que este tiempo de espera se puede optimizar haciendo que cada thread establezca una conexión distinta. La forma en que se recorre este grafo de documentos va a ser muy importante, ya que sabiendo que no se alcanzará a visitar por completo, va a ser necesario priorizar las páginas más relevantes.

Un crawler es el programa que ejecuta el web crawling, el cual en términos muy simplificados tiene la siguiente estructura:

1. URL frontier.
2. Módulo de fetching.
3. Módulo de parsing.

El punto de partida es una lista inicial de URLs a descargar llamada seed set, las cuales son añadidas al URL frontier. El frontier tiene la función de almacenar y ordenar por prioridad las URLs que serán descargadas. En general al hacer web crawling se descargan tantas páginas

[3]La página de las herramientas de Wikidata se encuentra en `https://www.wikidata.org/wiki/Wikidata:Tools`

que en la práctica es imposible obtenerlas todas, por lo que el frontier las jerarquiza siguiendo algunos parámetros. Las URLs se pueden ordenar mediante su utilidad estimada como el tráfico que reciben, la reputación de la página web, o su PageRank. El URL frontier le entrega al módulo de fetching la URL con mayor valor esperado para que la descargue. Luego el módulo de parsing se encarga de guardar el contenido del documento y todos los demás enlaces que encuentre. Estos nuevos enlaces se pasan al URL frontier para que actualice la lista de URLs y las ordene; luego se repite el proceso hasta que en teoría no se descubra ninguna página nueva. Al paralelizar, cada proceso ejecuta este loop manteniendo su propio URL frontier, pero asegurándose de que no se descarguen páginas repetidas.

Dentro de las consideraciones que debe proveer un crawler se encuentra el estar diseñado a prueba de trampas. Esto significa que debe ser capaz de detectar cuando ha entrado en un loop infinito descargando páginas en algún servidor y detenerse. Esto puede ser causado por un servidor malicioso especialmente diseñado para dejar atrapados a los web crawlers, aunque también de forma involuntaria en páginas web mal diseñadas.

Una de las limitaciones más importantes a la que deben regirse es seguir la políticas que regulan los web crawlers. Estas se refieren a que los crawlers no pueden acceder a los sitios las veces y el tiempo que quieran, sino que deben ser mesurados para no afectarlos negativamente y evitar un ataque involuntario de denegación de servicio por causa de excesivo tráfico. Para solucionar esto, al querer acceder a un mismo sitio, es necesario esperar un tiempo prudente de algunos segundos entre peticiones. Debido a lo anterior, el orden de la lista de URLs a descargar no puede ser totalmente arbitraria, ya que si URLs de un mismo sitio aparecen contiguas o muy juntas, el proceso se vuelve menos eficiente, ya que pierde tiempo esperando en vez de estar haciendo solicitudes a otros sitios. Para maximizar el número de descargas en un mismo período de tiempo, es trabajo del crawler considerar el tiempo que debe esperar entre solicitudes a un mismo sitio. Esto hace que el tiempo que tardará un crawler en descargar un grupo de páginas dependa en la diversidad de hosts distintos de la procedencia de las URLs, y cómo éste logre balancear las solicitudes de forma eficiente.

Otra política que los web crawlers deben cumplir es el Robots Exclusion Protocol. Este consiste en un archivo llamado *robots.txt* que un host puede dejar en el directorio raíz de una página. En él se pueden especificar reglas que van dirigidas a los crawlers para evitar que descarguen ciertas páginas o limitar número de consultas. Por ejemplo, para hacer que los crawlers ignoren algún directorio, hay que añadir la siguiente línea a robots.txt:

```
User-Agent: *
Disallow: /ejemplo/
```

El campo "User-Agent" corresponde al nombre del crawler y "Disallow" el directorio que se desea que no acceda. Por ejemplo si se quiere que el crawler de Google no acceda a ningún directorio, hay que añadir:

```
User-Agent: googlebot
Disallow: /
```

Además de esto se puede especificar el campo "Crawl-delay", que corresponde al número de segundos que un crawler debe esperar entre peticiones sucesivas. Por ejemplo si que quiere que los crawlers esperen al menos 10 segundos entre 2 consultas consecutivas, se debe añadir:

```
User-Agent: *
Crawl-delay: 10
```

Agregar instrucciones en el robots.txt ayuda a los servidores a disminuir el tráfico, y siempre va a depender de los crawlers si las siguen o no.

Antes de visitar cualquier página, los crawlers deben primero descargar el archivo robots.txt (en caso de que exista) para saber a qué archivos tienen acceso, luego pueden guardarlo en un caché para la próxima ocasión que visiten el mismo host.

2.5. Índices Invertidos

Una vez que se ha descargado una gran cantidad de documentos mediante web crawling, estos se almacenan tradicionalmente en un índice en donde para cada documento se guarda su contenido. Pero en el caso que se quiera implementar una búsqueda de texto, como es el caso de los motores de búsqueda, es necesario crear un índice invertido. Este índice se crea con el contenido de los documentos, en donde para cada palabra se mapea el documento en donde aparece. De esta manera se puede implementar eficientemente búsqueda de texto. Por ejemplo, se puede realizar una consulta compuesta de un conjunto de palabras en donde el resultado corresponde a los documentos en donde aparecen estas palabras, sin necesidad de buscarlas secuencialmente en cada uno de los documentos, lo cual sería computacionalmente costoso.

Esta manera de buscar documentos es efectiva pero básica. En aplicaciones reales se tienen miles de documentos y al hacer búsqueda de texto se requiere que los resultados sean acotados y relevantes a lo que se está buscando. Es por esto que existen muchos algoritmos para medir la relevancia de palabras en conjuntos de documentos. Uno de ellos es "Term Frequency - Inverse Term Frequency" o TF-IDF.

2.6. TF-IDF

Como el nombre lo sugiere, este método consiste en darle un valor de relevancia a las palabras de un documento en base a la cantidad de veces que aparece en él según su popularidad en todo el set de documentos. Un valor alto de TF-IDF significa que esa palabra es relevante en el documento que se encuentra. La idea es que palabras poco comunes que aparecen seguidamente en un documento tienden a tener mayores valores de TF-IDF, mientras que

palabras muy comunes como preposiciones y artículos tienden a tener menores valores, ya que aparecen en casi todos los documentos, y por lo tanto, no son especialmente relevantes para ningún documento en particular. Usando este valor se pueden implementar algoritmos para realizar búsquedas de relevancia de documentos. Por ejemplo, los documentos más relevantes dado un grupo de palabras serán los que maximizan la suma de valores de TF-IDF de los términos.

A pesar de que es un método relativamente simple, ha demostrado ser eficaz al hacer búsqueda de documentos relevantes para búsqueda de texto. Es el usado por defecto en la librería Apache Lucene™ que es un software para implementar motores de búsqueda. Sin embargo, en su forma más elemental, TF-IDF tiene algunas desventajas. Por ejemplo, no establece relaciones entre palabras, lo que se traduce en que sinónimos o plurales no son considerados como términos semejantes, en cambio, los interpreta como palabras sin conexión. Estos son problemas que son solucionados en versiones más complejas de TF-IDF.

2.6.1. Algoritmo de ranking de Apache Lucene

El algoritmo de ranking de relevancia por defecto de Lucene es el que se usa en la solución de este trabajo. Se basa en el modelo de espacio vectorial (MEV) de recuperación de información, en donde los pesos corresponden a valores TF-IDF. Lucene refina el valor MEV para mejorar la calidad de las búsquedas y usabilidad. La *fórmula conceptual de puntaje de Lucene* dada una consulta q y un documento d es la siguiente:

$$score(q,d) = coordFactor(q,d) \cdot qBoost(q) \cdot \frac{V(q) \cdot V(d)}{|V(q)|} \cdot dLenNorm(d) \cdot dBoost(d) \quad (2.1)$$

En donde:

- ***coordFactor(q, d)*** es un factor que puede usarse para entregar mayores puntajes a documentos que contengan mayor cantidad de términos de la consulta (en caso de consultas con más de un término).
- ***qBoost(q)*** puede usarse para que algunos términos en la consulta contribuyan más al puntaje de un documento por sobre otros.
- ***V(q)*** y ***V(d)*** corresponden a los vectores del MEV.
- ***dLenNorm(d)*** se utiliza para evitar perder información sobre el largo del documento al normalizarlo. Este es un factor de normalización que normaliza a un vector igual o mayor al unitario.
- ***dBoost(d)*** es un valor para representar que algunos documentos son más importantes que otros, lo cual puede ser definido por los usuarios al momento de indexar.

A partir de esta fórmula conceptual se deriva la *fórmula práctica de puntaje de Lucene* que es la siguiente:

$$score(q,d) = \frac{coord(q,d)}{qNorm(q)} \cdot \sum_{t\ en\ q} [TF(t\ en\ d) \cdot IDF(t)^2 \cdot tBoost() \cdot norm(t,d)] \tag{2.2}$$

En donde:

- ***coord(q,d)*** es el factor que aumenta el puntaje en base a cuántos términos están presentes en el documento, derivado de $coordFactor(q,d)$ de la fórmula conceptual.
- ***qNorm(q)*** es un factor de normalización para que puntajes de consultas diferentes sean comparables. No afecta el ranking de los documentos ya que todos son multiplicados por el mismo factor. Este valor se deriva del término $|V(q)|$ de la fórmula conceptual y se calcula mediante la norma euclidiana, es decir, la raíz cuadrada de la suma de los pesos de los términos de la consulta al cuadrado.
- ***TF(t en d)*** corresponde a la cantidad de veces que aparece el término en el documento. Por defecto el valor se calcula así $TF(t\ en\ d) = \sqrt{frecuencia}$.
- ***IDF(t)*** corresponde a qué tan raro es el término en el conjunto de documentos. Se calcula de la siguiente manera $IDF(t) = 1 + log(\frac{nDocs}{fDocs})$, en donde $nDocs$ es la cantidad de documentos y $fDocs$ es la cantidad de documentos en donde aparece el término. El valor $TF(t\ en\ d) \cdot IDF(t)^2$ de la fórmula práctica corresponde al término $V(q) \cdot V(d)$ de la fórmula conceptual.
- ***tBoost()*** es el factor para aumentar el valor que contribuye algún término en la consulta. Corresponde al término $qBoost(q)$ de la fórmula conceptual.
- ***norm(t,d)*** encapsula los factores $dLenNorm(d)$ y $dBoost(d)$ de la fórmula conceptual. Sirven para la normalización del largo de los documentos y para darle a algunos documentos mayor relevancia por sobre otros respectivamente.

Capítulo 3

Problema

En esta sección se presentará en detalle el problema abordado, junto con algunas estadísticas actuales para caracterizarlo y los desafíos que esto implica para la construcción de la solución.

3.1. Descripción

Wikidata exige que la información de las afirmaciones esté debidamente respaldada mediante una referencia, y ésta debe cumplir con ser confiable, estar actualizada y estar libre de sesgo. Dentro de lo que se acepta como fuente referenciable se encuentran por ejemplo:

- Publicaciones científicas
- Diarios, libros, revistas.
- Páginas Web

Las referencias se agregan a nivel de afirmación, la cual puede puede tener 1 o más. Una referencia es más que solo un par propiedad-valor, consiste en un grupo de calificadores, los cuales juntos conforman una referencia. Dentro de una referencia el calificador más importante corresponde al que indica el recurso donde la información es respaldada. En el caso de usar una URL externa se usa la propiedad "URL de la referencia" (P854), donde el valor es la URL. En caso de querer validar un hecho usando un elemento de Wikidata, se usa la propiedad "Afirmado en" (P248), donde el valor corresponde al elemento. Wikidata sugiere usar preferentemente referencias internas, ya que son básicamente elementos que representan organizaciones o páginas web, y esto permite una mayor conexión de datos. Pero si no existe un elemento que represente una página web, no hay problema con usar la URL como referencia. También se sugiere agregar otros calificadores para darle información contextual a la referencia, como por ejemplo la fecha en que se agregó. Hay que mencionar que no todas las

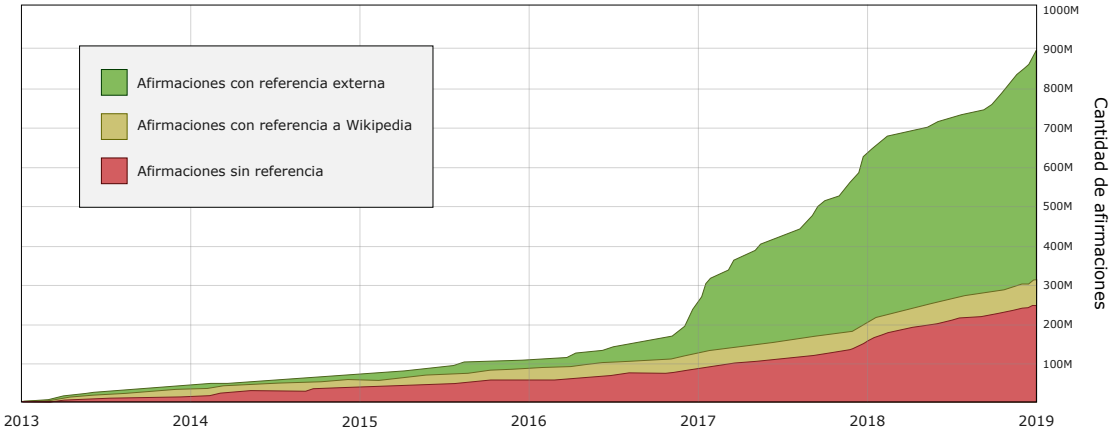

Figura 3.1: Distribución del tipo de referencia que presentan las afirmaciones en Wikidata a lo largo del tiempo.

afirmaciones necesitan tener una referencia, aunque este grupo no representa la mayoría de las afirmaciones. Las que están exentas son las que correspondan a alguno de estos casos:

1. Afirmaciones que son indisputables (e.g. la Tierra es un planeta).
2. Cuando una afirmación no es una propiedad del elemento, sino que lo está conectando con otra fuente de información (e.g. un id en una base de datos externa).
3. Cuando la fuente de una afirmación se halla en el elemento mismo (e.g. el autor de un libro cuando esta información está en el libro mismo).

En Wikidata existen aproximadamente 70 millones de elementos y 890 millones de afirmaciones. En la práctica no todas las afirmaciones están validadas. Actualmente hay 246 millones de afirmaciones sin referencia. Además de esto, hay un grupo de 60 millones de hechos que cuentan con referencia pero ésta es un enlace a una página de Wikipedia, como es el caso de muchas afirmaciones añadidas por bots que importan información desde Wikipedia, estas referencias no corresponden a fuentes válidas ya que se sugiere siempre referenciar una fuente primaria directamente, por lo que se pueden considerar como afirmaciones sin referencia. En total, si consideramos ambos grupos, existen algo así como 306 millones de afirmaciones sin referencia, lo cual corresponde al 34 % del total (ver figura 3.1). Además de lo anterior, en un estudio del año 2017 [13], vieron que solo un 61 % de las referencias de los hechos en Wikidata eran relevantes y autoritativas, el resto eran enlaces que ya no funcionaban o que apuntaban a otra parte.

Esto es un problema para Wikidata ya que para cualquier fuente de datos secundaria es muy importante la procedencia de la información. Si los datos tienen referencias que no son relevantes o simplemente no tienen, estos pierden valor, ya que dejan de ser confiables para cualquier organización o persona que quiera hacer uso de ellos. Tanto es así que en julio del año 2018, la aplicación de voz Siri de Apple anunciaba erróneamente la muerte de Stan Lee (Stan Lee murió en noviembre de ese año), debido a que automáticamente importaba

datos de Wikidata, y este dato había sido ingresado sin referencia con una fecha equivocada. Eventualmente Siri corrigió esto, pero no quita que pueda volver a pasar con alguna otra información. Por esto es que Wikidata exige añadir referencias confiables a su información, y para ayudar a sus colaboradores a encontrarlas propone principalmente tres estrategias[1], estas son:

1. Para los elementos que tienen su contraparte en Wikipedia, buscar en la sección de "Referencias" enlaces que puedan ser relevantes para lo que se quiere validar.
2. Buscar en recursos como: Google, Google News, Google Newspapers, Google Books, Google Scholar y JSTOR[2].
3. Buscar en la librería de Wikipedia que contiene una lista de fuentes autoritativas[3].

En términos de herramientas, en la página oficial de Wikidata se menciona que "se está trabajando en desarrollar una herramienta para buscar referencias" y que mientras tanto "existen algunos programas para los colaboradores de Wikidata que ayudan a buscar y añadir referencias", sin embargo, no se especifica ni se hace referencia a ninguno de estos programas. Lo único que se menciona es una página especial de fuentes de libros para buscar mediante su identificador ISBN. Ingresando en esta página el ISBN de un libro, ésta provee enlaces a catálogos de bibliotecas, bases de datos, y otros sitios con información acerca del libro. En la práctica está no es una buena solución, ya que conlleva mucho trabajo por parte de un colaborador, considerando además que en muchos casos es probable que los libros no estén disponibles.

Sin embargo, aún así existe una gran cantidad de hechos sin referencia. Una de las posibles razones de por qué sucede esto es que encontrarlas es difícil. Las formas que propone Wikidata para buscar referencias contemplan un trabajo manual que puede ser tedioso y puede tomar mucho tiempo. Por esta razón, colaboradores podrían preferir dejar hechos simplemente sin referencia. Además, Wikidata cuenta con bots que añaden afirmaciones de forma automática, y estos no incluyen referencias cuando ingresan información. Wikidata se vería beneficiada si contara con algún mecanismo automatizado que ayudara los colaboradores a encontrar referencias a sus afirmaciones de forma más simple.

3.2. Situación Actual

La primera estrategia que sugiere Wikidata para encontrar referencias consiste en buscar en las páginas de Wikipedia de los elementos de Wikidata. Esta es una buena opción de solución ya que la sección de referencias de los artículos de Wikipedia en sí misma es una

[1]Los métodos para encontrar referencias están en `https://www.wikidata.org/wiki/Wikidata:Verifiability`

[2]JSTOR es un repositorio digital de libros, revistas científicas y fuentes primarias `www.jstor.org`

[3]La lista de fuentes autoritativas se encuentra en `https://en.wikipedia.org/wiki/Wikipedia:List_of_free_online_resources`

buena fuente de referencias autoritativas, ya que Wikipedia exige que su información provenga de fuentes confiables y es incluso más estricta que Wikidata en eso, por lo que es buena idea buscar allí. Además los elementos de Wikidata comparten mucha información con los respectivos artículos de Wikipedia, por lo que es natural pensar que los mismos enlaces se puedan reutilizar para validar información en ambas plataformas. Sin embargo, en un estudio del año 2017 [14], analizaron las referencias externas Wikipedia y Wikidata y encontraron que existían una baja reutilización de referencias en los artículos y sus respectivos elementos. Específicamente, solo el 1,7 % de los elementos tenía 1 ó más referencias en común con su artículo de Wikipedia, y aquellas que se reutilizaban entre artículo-elemento correspondían solo al 0,85 %. Para este trabajo se hizo el mismo análisis en un conjunto de 5000 elementos elegidos al azar y se obtuvieron resultados similares: solo el 1,4 % de los elementos contenía al menos una referencia en común con su respectiva página en Wikipedia y el 0,42 % de las referencias se compartían entre elemento-artículo. Estos resultados indican que las referencias de Wikipedia no se están reutilizando para los elementos en Wikidata, lo cual es un poco sorprendente, ya que esto es precisamente lo primero que propone Wikidata para encontrar referencias.

Para que esta solución que propone Wikidata sea efectiva requiere principalmente:

1. Que el elemento de Wikidata al que le estamos buscando referencias tenga un respectivo artículo en Wikipedia.
2. Que el artículo contenga una cantidad razonable de referencias y que estas sean variadas en su contenido.

Y puede que esto en general no esté ocurriendo. La versión en inglés de Wikipedia tiene aproximadamente 5.900.000 artículos, y todos tienen un respectivo elemento de Wikidata. Pero Wikidata tiene 70 millones de elementos, por lo que hay 64.000.000 para los cuales encontrar referencias no será tan efectivo. Afortunadamente estos son, en general, los menos importantes, a los cuales no se le querrá añadir referencia preferentemente. Para este trabajo, considerando aquellos elementos que sí tienen contraparte en Wikipedia, de un total de 5.461.400 artículos se extrajeron 32.329.988 referencias (contando repeticiones), obteniendo en promedio 5,9 ($\pm$17.9) referencias por elemento. Los elementos que cuentan con más referencias son aquellos que tienen identificadores menores, ya que son los primeros que se añadieron y por lo tanto los más populares, como países, personas famosas y conceptos generales.

La distribución de referencias por elemento se muestra en la figura 3.2. Esta indica que va a ser mucho más fácil encontrar referencias para los primeros elementos, y rápidamente va a empezar a costar más para los elementos posteriores, ya que incluso muchos no tienen ninguna referencia en sus páginas de Wikipedia.

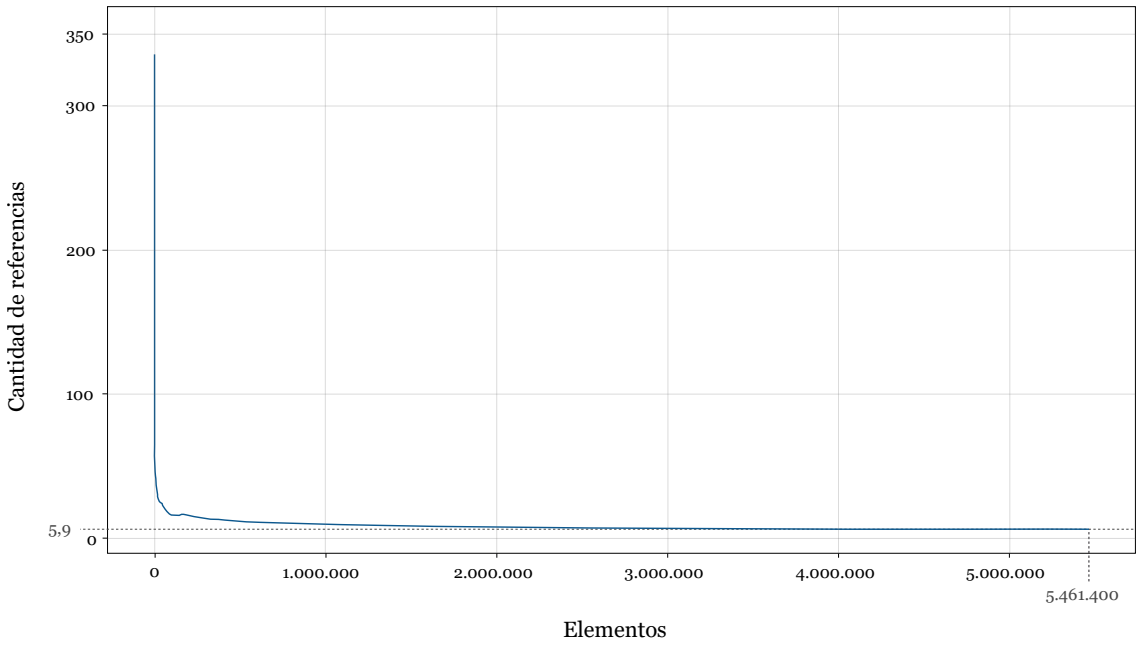

Figura 3.2: Promedio acumulado de referencias por elemento, para los elementos ordenados por identificador de menor a mayor.

3.3. Desafíos

Hay 4 aspectos que principalmente afectarán a la calidad y usabilidad de la solución, estos son:

1. **Elementos con muy pocas o sin referencias**: Dependiendo del elemento, hay algunos que no tendrán un conjunto de referencias disponibles para buscar fuentes, por ejemplo elementos poco comunes con páginas de Wikipedia sin referencias, o simplemente aquellos que no tienen un respectivo artículo. Para este tipo de elemento se debería también poder buscar referencias, aunque se puede suponer que en estos casos los resultados no sean los óptimos.

2. **Relevancia de las referencias retornadas por afirmación**: Una vez que se tenga un conjunto de referencias descargadas e indexadas para un cierto elemento, para cada afirmación se tendrán documentos que la validen y otros que no serán relevantes. La solución debe retornar para cada afirmación sólo aquellos documentos relevantes e intentar no mostrar los que sean irrelevantes. Esto es particularmente difícil dado que muchas veces un documento confirma una afirmación de forma indirecta a través de un proceso deductivo, que para las personas es fácil identificar pero no para programas simples. En otras ocasiones en las páginas se utilizan sinónimos o plurales de las palabras usadas para las afirmaciones, lo cual también presenta un problema para una solución que determine referencias mediante comparaciones de palabras exactas.

3. **Mostrar contenido de la página en donde se validan las afirmaciones**: Lo ideal es que los colaboradores no tengan que visitar cada página para verificar si en ellas se validan o no las afirmaciones. Si en un documento se valida una afirmación, y

esta muestra el extracto donde se dice dicha información, se puede ahorrar trabajo al colaborador resultado en un sistema más usable.

Capítulo 4

Solución

Primero se mostrará la arquitectura general de la solución, para luego detallar las alternativas que existen para los distintos procesos que conlleva. Estos son: la forma en que se obtienen los datos de los elementos, las dos opciones online y offline en que se puede usar la herramienta, cómo se crea el índice de documentos y las opciones de rangos de búsqueda. Al final se mostrará un ejemplo de uso del sistema.

4.1. Descripción

La idea general de la solución consiste en una automatización del primer método que sugiere Wikidata para buscar referencias. Esto es, dado algún elemento, extraer y descargar las referencias de su página de Wikipedia para luego buscar en ellas referencias para las afirmaciones. La arquitectura consta principalmente de 2 componentes: una aplicación web y un índice de documentos construido con las referencias de las páginas de Wikipedia. Desde la aplicación web se ejecuta: la consulta de elementos junto con sus afirmaciones a Wikidata, y la búsqueda de documentos al índice. El proceso completo enumerado para buscar referencias se muestra en la figura 4.1.

4.2. Obtener Datos de Elementos

El proceso inicia buscando un elemento por su identificador al que se desea buscar referencias (punto **1** de la figura 4.1), para lo cual es necesario conocer previamente el identificador del elemento. Se hace una consulta al endpoint SPARQL de la API de Wikidata para obtener su nombre, descripción y lista de afirmaciones (punto **2** de la figura 4.1). Esto se hace cada vez que se busca un elemento, y a pesar de que podría optimizarse pre-cargando estos datos en un caché, se tiene la ventaja que se obtiene siempre la información más actualizada de Wikidata.

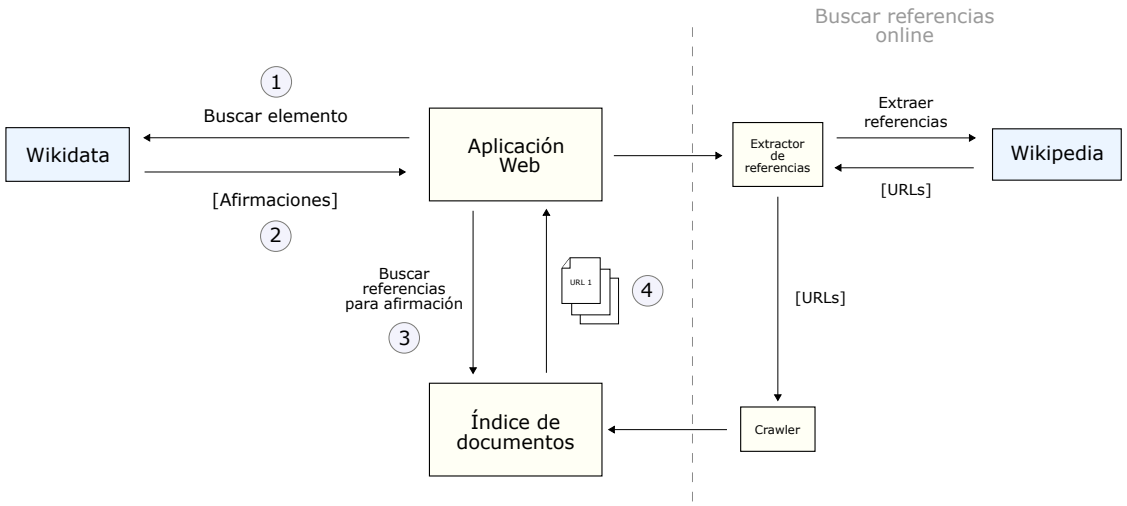

Figura 4.1: Arquitectura simplificada de la solución, en donde se muestran los 2 componentes principales y el proceso para buscar referencias a una afirmación.

4.3. Referencias del Índice

Una vez que se tiene la lista de afirmaciones del elemento se puede realizar la búsqueda de referencias. Para esto identificamos dos opciones: offline y online, las cuales determinarán entre otras cosas la cantidad de documentos disponibles en el índice para la búsqueda.

4.3.1. Opción offline

Dada una afirmación seleccionada por el usuario, esta alternativa se basa en usar solo los documentos previamente descargados que existen en el índice para buscar referencias, sin obtener nuevas páginas ni actualizar antiguas. Esta opción contempla la descarga de referencias "en bulto" en un proceso anterior, para así tenerlas indexadas antes de que el usuario haga su petición.

4.3.2. Opción online

Esta opción consiste en actualizar el índice con las referencias que existen en la página de Wikipedia del elemento antes de buscar referencias. Este proceso es: visitar y extraer las referencias del artículo de Wikipedia por medio del extractor de referencias, los enlaces se pasan al web crawler para que las descargue e indexe, bajando solo aquellas páginas que son necesarias descargar, es decir, que no existan previamente en el índice. Cada vez que se agregan documentos, estos quedan disponibles para la próxima vez que se busque referencias para el mismo elemento. La implementación del extractor de referencias corresponde a un script en Python usando la librería Beautiful Soup, mientras que para descargar e indexar

los documentos se usa Apache Nutch™.

4.3.3. Comparación de opciones: offline vs online

Ambas alternativas tienen ventajas y desventajas. Por una parte la opción offline permite realizar la búsqueda de referencias de forma casi inmediata, pero para que dé buenos resultados depende de que los documentos que existan en el índice sean relevantes al elemento que se está buscando referencias. En cambio la alternativa online asegura de alguna manera las condiciones óptimas para buscar referencias, ya que descarga referencias relevantes al elemento y también las últimas páginas disponibles. Sin embargo, tiene un costo al tener que esperar descargar las referencias, lo cual en general es lento y es peor entre más referencias se tenga que descargar. Lo ideal sería tener todas las referencias de Wikipedia descargadas con antelación, para así usar solo la versión offline de la aplicación y no tener que esperar ninguna descarga. Como en la práctica no se tienen todas descargadas, la versión online pretende ofrecer dar mejores resultados a costo de tener que esperar. El problema que se vio con la opción online es que, más allá de cuántas referencias se descarguen, toma mucho tiempo. Esto se debe a que el proceso contempla muchos pasos: preparar el crawler, inyectar URLs, descargarlas, parsearlas, indexarlas, detectar páginas iniciales cuando se indexan de redirecciones, emparejar estas páginas con el elemento de donde se obtuvieron, actualizar finalmente las páginas indexadas. Esto implica que los usuarios deben esperar varios minutos al querer descargar referencias para casi cualquier elemento de forma online. Esta espera resulta en que esta opción no es usable, por lo que se decidió no hacerla parte de la solución. En cambio, solo estarán disponibles los documentos que se descargaron al índice con antelación.

4.3.4. Índice de documentos

Para crear el índice primero se obtuvo un *dump* de Wikidata para tener un mapeo de elementos de Wikidata a sus respectivos artículos (en la versión en inglés) de Wikipedia, obteniendo una lista de URLs de todas las páginas de Wikipedia. A partir de esta lista de URLs se extrajeron los enlaces de la sección "Referencias" de cada artículo usando un dump de Wikipedia (del año 2018). Con esto se obtuvo principalmente dos cosas: un mapeo de todos los elemento de Wikidata a las respectivas URLs de las referencias obtenidas de Wikipedia, esto para saber de qué artículo provino cada enlace; y a la vez una lista de todas estas referencias.

Descargar e indexación de referencias

Esta lista de enlaces se pasaron al web crawler para que las descargue e indexe en tandas de 50.000 páginas cada vez, configurado para que siga las redirecciones, no visite otros enlaces que encuentre en las páginas originales y espere 5 segundos por defecto entre peticiones a un mismo sitio, entre otras opciones. Cada documento que el crawler indexa en el índice contiene

```
{
  "tstamp":"2019-09-26T22:35:59.773Z",
  "digest":"e00712b4d09b9742e13bce0f699eff41",
  "host":"news.bbc.co.uk",
  "boost":1.0,
  "id":"http://news.bbc.co.uk/2/hi/americas/country_profiles/1222764.stm",
  "title":"BBC News - Chile country profile",
  "url":"http://news.bbc.co.uk/2/hi/americas/country_profiles/1222764.stm",
  "Q":[298],
  "_version_":1647144810188898304,
  "content":"BBC News - Chile country profile\nBritish Broadcasting Corporation Home\nAccessibility links\nSkip to content\nSkip to
}
```


Figura 4.2: Ejemplo de un documento y sus campos como se muestra en la interfaz web de Solr™.

los siguientes campos:

- "**tstamp**": La fecha y hora en que se descargó el documento.
- "**digest**": Un string que representa un hash criptográfico del documento.
- "**host**": El sitio de donde se obtuvo el documento.
- "**boost**": Un valor que puede ser usado para aumentar o disminuir la prioridad de los documentos al momento de rankearlos en una búsqueda.
- "**url**": La URL del documento, en caso de ser un redirección esta URL corresponde a la URL final.
- "**id**": El identificador único del documento que en este caso es igual a la url.
- "**title**": El título de la página web.
- "**Q**": Una lista de identificadores de Wikidata en donde este documento aparece.
- "**_version_**": Un número que cambia cada vez que el documento se actualiza.
- "**content**": El contenido de la página web.

El índice de documentos corresponde a un servidor Apache Solr™, el cual junto con el crawler Apache Nutch™ son dos herramientas que fueron diseñadas especialmente para que la integración de ambas sea sencilla. En la figura 4.2 se muestra un ejemplo de un documento extraído del artículo de Chile, como se ve desde la interfaz gráfica de Solr™.

4.4. Rango de Búsqueda

La búsqueda de referencias se puede realizar entre dos rangos o conjuntos de documentos, los cuales son: aquellos obtenidos de la página de Wikipedia del elemento y todos los documentos del índice.

Limitar búsqueda a referencias del artículo de Wikipedia

La opción por defecto consiste en buscar sólo en aquellos documentos que aparecen en la página de Wikipedia del elemento, ya que se asume que estas páginas están relacionadas de forma cercana con el elemento.

No limitar búsqueda

La segunda opción no limita el rango de búsqueda de documentos, sino que se busca en todos los documentos disponibles en el índice.

4.4.1. Comparación de Opciones de Rangos de Búsqueda

La mejor opción para asegurar la relevancia de las referencias sugeridas es la que limita la búsqueda a los documentos que aparecen en el artículo de Wikipedia. En general casi todas estas páginas están relacionadas con el elemento y son usadas para validar información del elemento en Wikipedia, por lo que es razonable pensar que puedan ser usadas para validar información esta vez para hechos de Wikidata. Los artículos de Wikipedia tienen cantidad de referencias que van de ninguna a varios cientos, por lo que esta opción da un buen balance de documentos relevantes en donde buscar y a la vez acotados. Sin embargo, no siempre va a ser viable usar esta opción, por ejemplo en el caso que el artículo de Wikipedia del elemento no tenga ninguna referencia (o tenga muy pocas) o las referencias aún no han sido descargadas al índice. En estos casos va a ser necesario usar la alternativa de extender la búsqueda a todos los documentos del índice. Hacer esto aumenta significativamente la diversidad de páginas en donde se busca, pero a la vez tiene un efecto negativo al añadir muchos documentos que no tienen relación con el elemento a los posibles resultados. Usar la opción de extender la búsqueda a todos los documentos no necesariamente debe usarse en los casos mencionados, sino que puede ser usada libremente en cualquier situación para intentar dar con los mejores resultados posibles.

4.5. Términos de Búsqueda

La búsqueda de documentos o referencias (punto **3** de la figura 4.1) se realiza usando el algoritmo de búsqueda por defecto de Lucene, mencionado en la sección 2.6.1, sobre el contenido de las páginas.

El término de consulta por defecto usado se genera concatenando el nombre de la propiedad, el valor de la propiedad y el nombre del elemento al cual pertenece el hecho al cual se está buscando referencias. Cada término sin necesidad que alguna aparezca obligatoriamente en los resultados, de la siguiente manera:

Término de consulta = <nombre elemento> OR <propiedad> OR <valor>

De esta manera la búsqueda no es tan restrictiva, y permite encontrar documentos relacionados con los términos pero que en muchos casos no contienen todos los términos. En caso de que el nombre del elemento, de la propiedad o del valor esté a su vez compuesto de varias palabras, estas también se concatenan mediante cláusulas OR, para evitar que documentos queden fuera de los resultados debido a reglas demasiado estrictas.

El algoritmo de Solr que utiliza el método TF-IDF procede mediante la búsqueda de coincidencias exactas de términos, por lo que puede pasar que un documento valide una afirmación pero no se detecte, debido a que las palabras usadas en la consulta y en el documento no eran exactamente las mismas. Por ejemplo, si queremos validar la siguiente afirmación de Wikidata:

`‘‘Universidad de Chile’’ (Q232141) ⇒ ‘‘Incepción’’ (P571) ⇒ 1842`

Y buscamos en un documento que lo menciona de esta manera:

`‘‘La Universidad de Chile fue fundada en 1842’’`

En este caso no se detectará como documento válido porque la búsqueda no hace la coincidencia de términos sinónimos, ya que no se usa la misma palabra para “incepción”, se usan “fundada en”. En este caso sí se debió haber reconocido este extracto como válido. Para intentar solucionar este problema se usarán las etiquetas alternativas de las entidades de Wikidata. En el ejemplo anterior “Universidad de Chile” e “incepción” corresponden a las etiquetas preferenciales, pero cada entidad de Wikidata provee además otros nombres alternativos llamadas “etiquetas alternativas” que son equivalentes a la etiqueta preferencial. Si en el ejemplo anterior se usan las etiquetas alternativas para la propiedad “incepción”, sí detecta el documento como coincidencia válida, ya que una de las etiquetas alternativas de la propiedad “incepción” es “fecha de fundación”. Sin embargo, a pesar de que añadir etiquetas alternativas puede ayudar a solucionar este problema, hay que considerar que también tiene un efecto negativo. Usar muchas etiquetas adicionales extiende los resultados de gran forma, lo cual no necesariamente va a ser beneficioso, ya que se añaden posiblemente documentos irrelevantes a los resultados. Por esta razón, haciendo uso de las etiquetas que provee Wikidata, se consideraron 4 opciones distintas para los términos de búsqueda, estas son:

- **Opción 1**: La etiqueta preferencial del nombre del elemento, de la propiedad y del valor.
- **Opción 2**: La etiqueta preferencial del elemento, de la propiedad, del valor y las etiquetas alternativas de la propiedad.

- **Opción 3**: La etiqueta preferencial y etiquetas alternativas del elemento, de la propiedad y del valor.
- **Opción 4**: Los mismos términos de la opción 3 pero con la restricción adicional que al menos una de las denominaciones del elemento, de la propiedad y del valor deben aparecer en el documento.

En el caso que el valor corresponda a un literal, se usa simplemente el valor del literal como string. La opción 1 es por defecto la usada por el programa, y el usuario puede elegir entre usar esta alternativa o la tercera opción, la razón para esto se discutirá en la siguiente sección. Los nombres alternativos se obtienen mediante una consulta a la API de Wikidata de igual forma como se hace con la lista de afirmaciones.

Por ejemplo, para el hecho:

‘‘Chile’’ (Q298) ⇒ ‘‘official language’’ (P37) ⇒ ‘‘Spanish’’ (Q1321)

Las etiquetas y términos de búsqueda que conforman el *query string* de cada opción anterior serían los siguientes:

Opción 1

Etiquetas		
Chile	Official language	Spanish

Query String
“Chile **OR** (Official, language) **OR** Spanish”

Opción 2

Etiquetas		
Chile	Official language Language spoken Spoken in Speaking	Spanish

Query String
"Chile
OR
(Official language, Language spoken, Spoken in, Speaking)
OR
Spanish"

Opción 3

Etiquetas		
Chile	Official language	Spanish
República de Chile	Language spoken	Castilian
Republic of Chile	Spoken in	Spanish language
CHL	Speaking	Español

Query String
"(Chile, República de Chile, Republic of Chile, CHL)
OR
(Official language, Language spoken, Spoken in, Speaking)
OR
(Spanish, Castilian, Spanish language, Español)"

Opción 4

Etiquetas		
Chile	Official language	Spanish
República de Chile	Language spoken	Castilian
Republic of Chile	Spoken in	Spanish language
CHL	Speaking	Español

Query String
"(Chile, República de Chile, Republic of Chile, CHL)
AND
(Official language, Language spoken, Spoken in, Speaking)
AND
(Spanish, Castilian, Spanish language, Español)"

4.6. Documentos Retornados

El resultado de realizar la búsqueda sobre el índice entrega 3 documentos (punto **4** en la figura 4.1), independientemente de cuál opción de búsqueda se utilice. Cada resultado contiene la URL y un extracto del contenido de la página en donde se encuentran coincidencias de los términos de búsqueda. La idea es que mirando este extracto un colaborador pueda decidir si esa página valida la afirmación o no sin tener que visitar la URL cada vez. El extracto que se entrega corresponde a una de las opciones que ofrece Solr™ para enriquecer los resultados de la búsqueda.

4.7. Ejemplo de uso

A continuación se dará un ejemplo de uso de la aplicación, en donde se mostrarán los pasos para buscar referencias a una afirmación de un elemento usando la versión offline de la aplicación. Se usará "Chile como ejemplo de elemento y "capital Santiago" como ejemplo de afirmación.

La pantalla inicial del programa muestra solo un input para buscar elementos, en él buscamos el identificador "298" que corresponde al país "Chile" (figura 4.3).

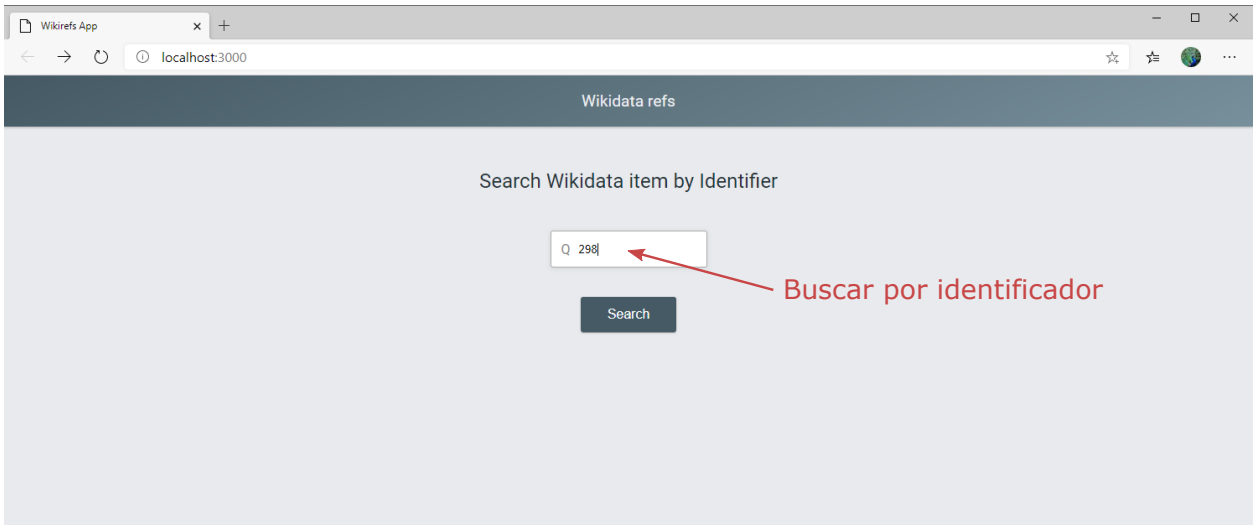

Figura 4.3: Pantalla inicial para buscar identificadores.

Al realizar la búsqueda y luego de unos segundos se obtienen los resultados y se muestran el nombre del elemento, la descripción y la lista de afirmaciones (figura 4.4).

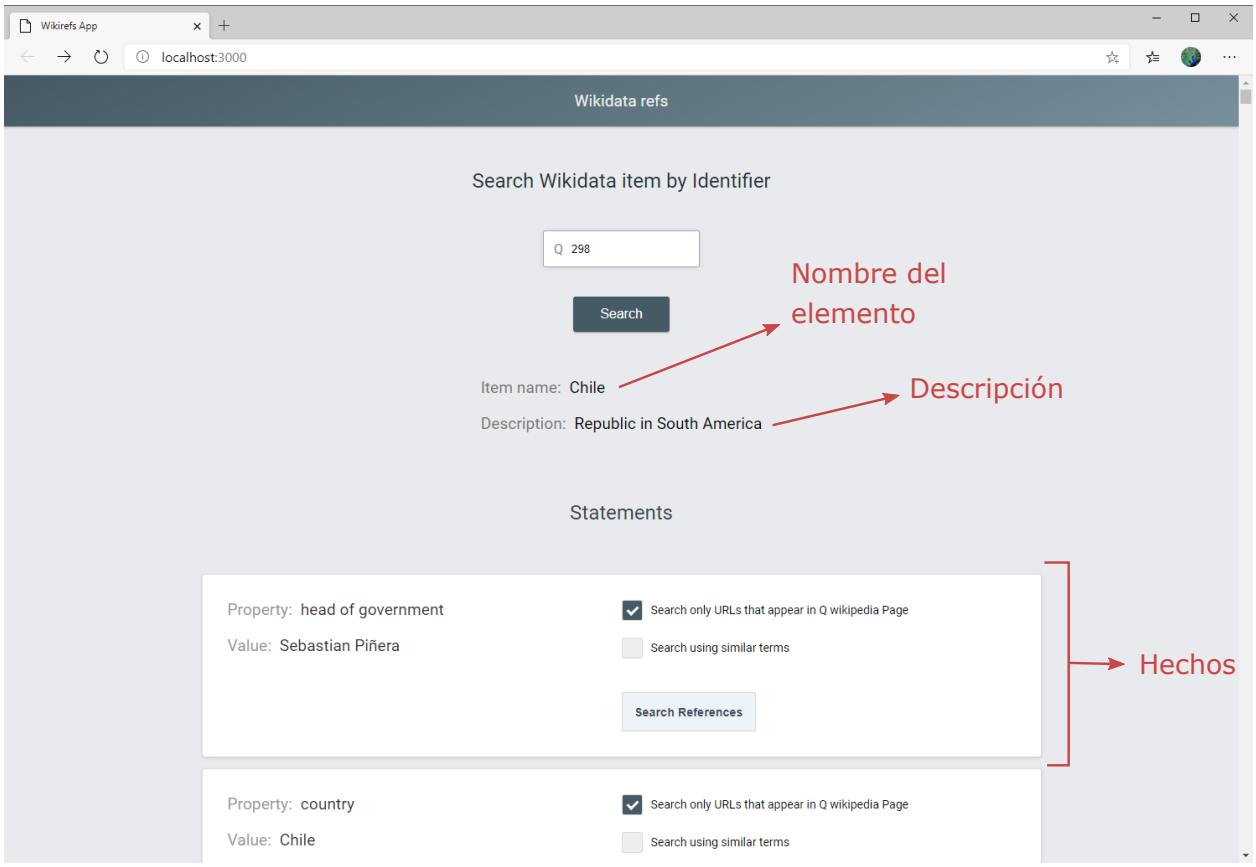

Figura 4.4: Vista del programa luego de buscar un elemento, en donde se muestran el nombre del elemento, descripción y lista de afirmaciones.

Identificamos la afirmación correspondiente “capital” y buscamos referencias (sin cambiar ninguna opción), obteniéndose 3 resultados. En este ejemplo se puede ver que la primera página retornada es un buen candidato a validar el hecho correctamente, ya que analizando la URL se puede observar que corresponde a la página de Chile de un diccionario; y en el extracto se menciona que Santiago es la capital y ciudad más grande (figura 4.5).

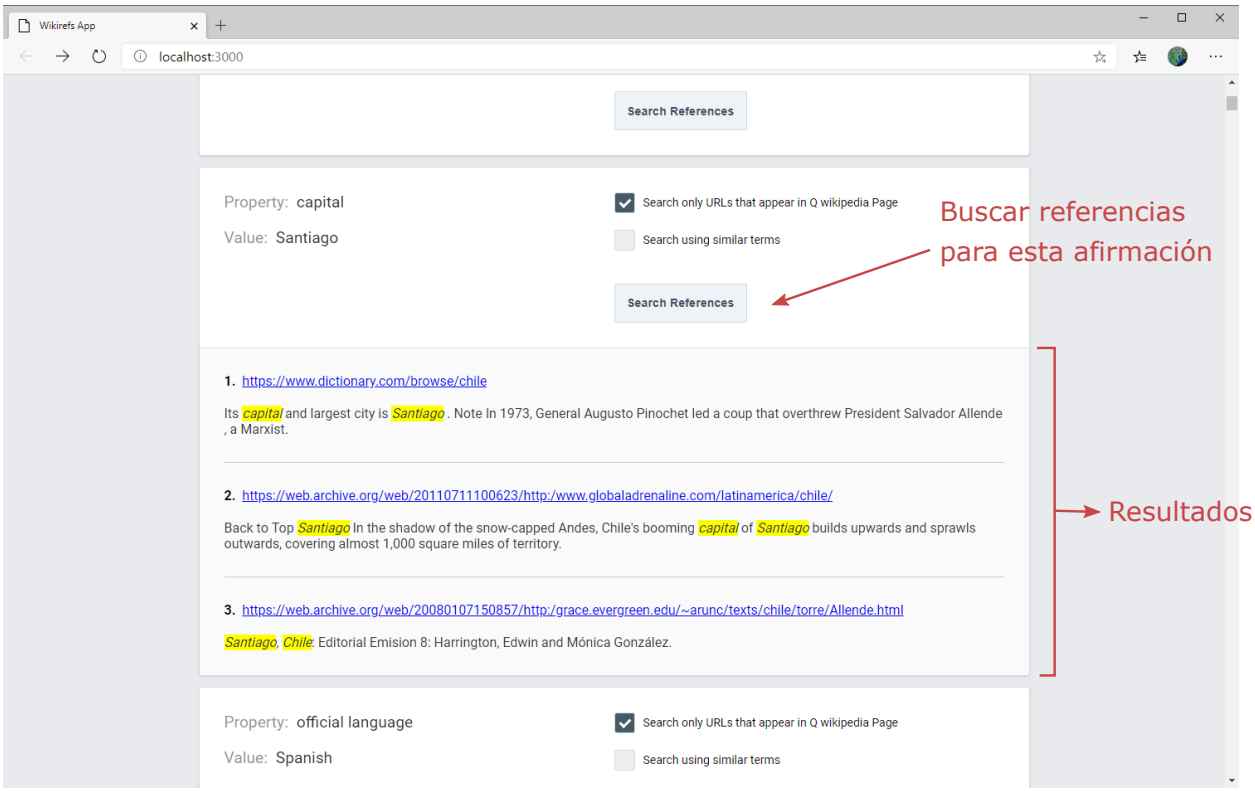

Figura 4.5: Vista de una afirmación luego de buscar referencias. En ella se pueden ver los 3 resultados retornados por el sistema.

Cambiando las opciones de búsqueda se obtienen distintos resultados. Para la misma afirmación, buscando en todo el índice y con términos de búsqueda alternativos, se puede ver que en este caso no se obtienen resultados relevantes (figura 4.6).

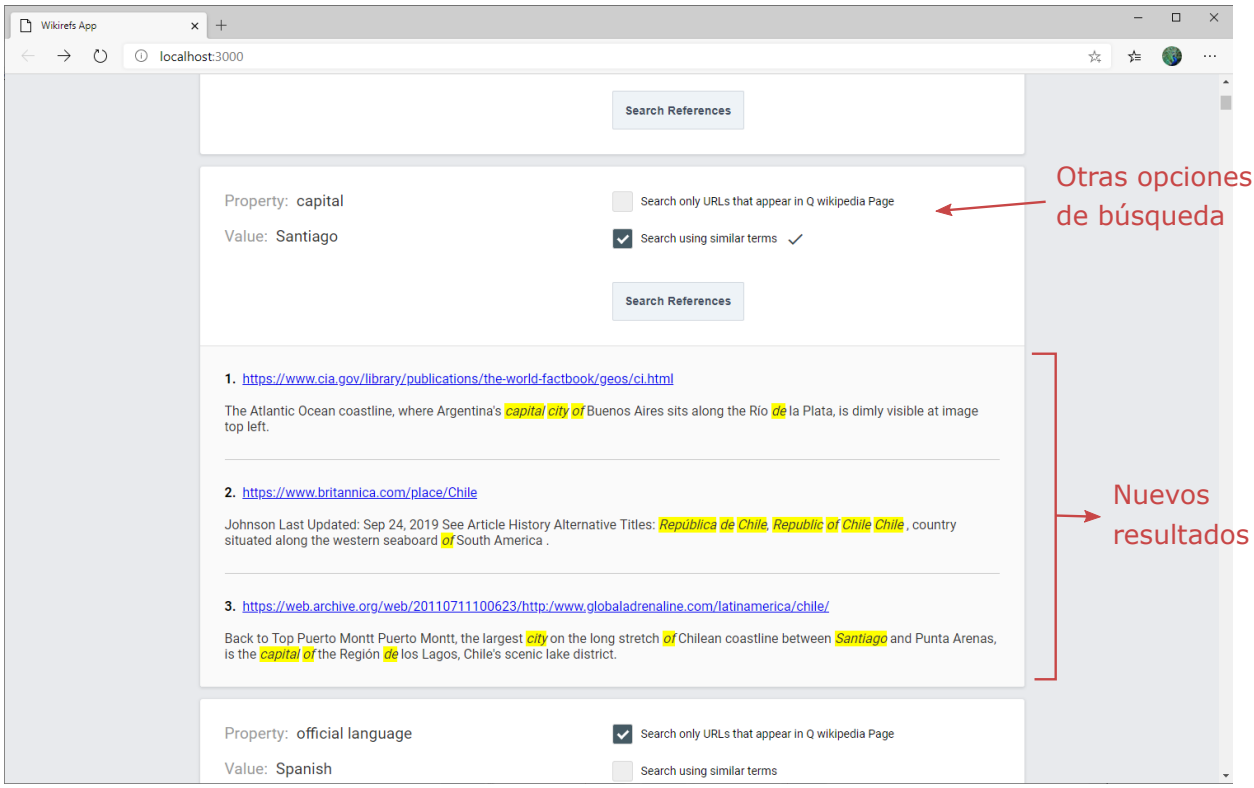

Figura 4.6: Vista de una afirmación al realizar la búsqueda de referencias con distintas opciones.

Capítulo 5

Validación

La validación consistió en medir principalmente dos cosas: el rendimiento en general de la solución y las distintas opciones de búsqueda mencionadas en la sección 4.5. Para poder medir el rendimiento de los resultados se compararon con una línea base de resultados aleatorios.

5.1. Datos

5.1.1. Crawl Completo

En una primera etapa se extrajeron todas las referencias de las páginas de Wikipedia, obteniéndose 32.329.989 URLs de un total de 5.461.401 elementos. Luego de realizar un filtro inicial para eliminar enlaces repetidos y mal formados, se obtuvieron 23.036.318 URLs. Esta lista se inyectó al crawler, el cual a su vez realizó otro filtro el cual finalmente generó en total 17.781.974 URLs para que sean descargadas.

El proceso del crawl inició en agosto de 2019 y se detuvo en diciembre de 2019 (con algunas pausas en este último mes). La figura 5.1 muestra el progreso de la cantidad de referencias descargadas e indexadas a lo largo de los meses que se mantuvo el crawl. Esta muestra una velocidad relativamente pareja de documentos descargados por día, entre agosto y septiembre el promedio fue de 19.172 mientras que entre noviembre y diciembre de 34.344. Este aumento se debe probablemente a que en noviembre se redujo el límite de peso de los documentos que el crawler descargue.

En total se lograron descargar 2.591.291 URLs de las cuales se indexaron 2.475.461. Este es un número bajo en cuanto a las expectativas que se tenían. La lenta velocidad de descarga se debe probablemente a los 5 segundos de espera entre peticiones a un mismo sitio, y a que en las tandas de descargas se generaban cuellos de botella. Estos ocurrían cuando quedaban muchos enlaces restantes que descargar pero de pocos sitios distintos, bajando la eficiencia al generar muchos threads ociosos. La figura 5.2 muestra el estado final de los enlaces agregados

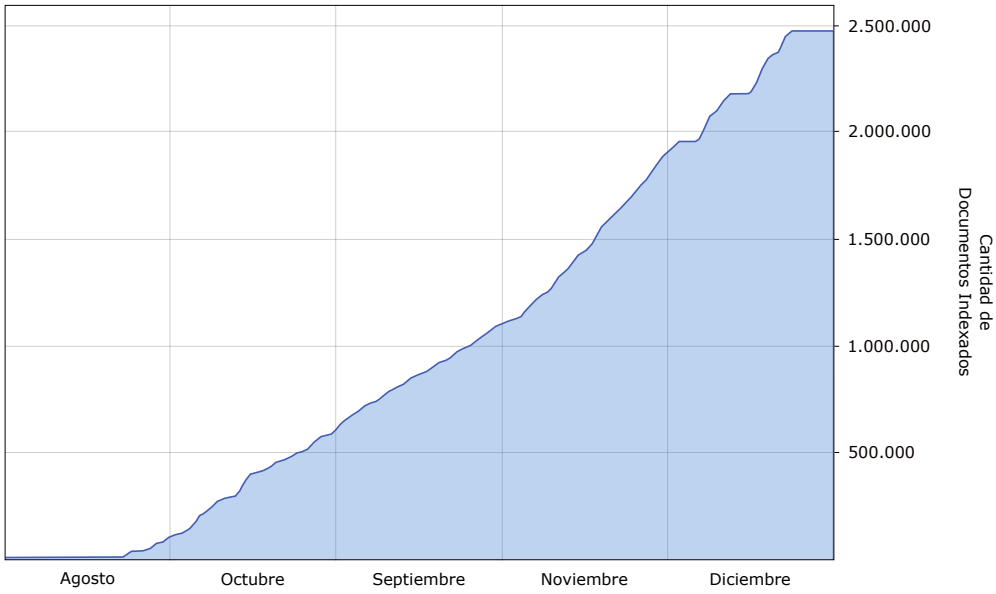

Figura 5.1: Progreso de documentos descargados e indexados en el tiempo por el crawler.

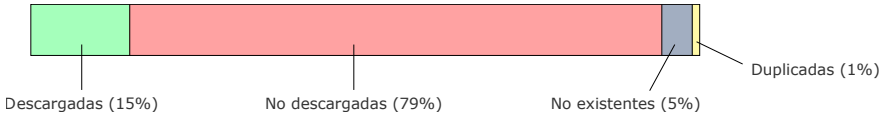

Figura 5.2: Estado final de los enlaces de la base de datos del crawler.

al crawler inicialmente, en donde se ve que las referencias descargadas corresponden a solo el 15 % del total.

5.1.2. Referencias Indexadas

En el tiempo que se ejecutó el crawl se indexó aproximadamente el 96 % de las páginas descargadas, o el 14 % (2.475.461) del total. Esto se puede atribuir a problemas con parsear algunos documentos debido a sus contenidos. Luego se realizó el proceso de asignar a cada documento el o los identificadores de Wikidata de donde se obtuvo ese documento. Se consiguió emparejar 2.058.896 con sus respectivas páginas de Wikipedia de donde se obtuvieron, el cual representa el 83 % del total de documentos indexados. Los que no se lograron emparejar fueron documentos que no se pudo identificar su origen. Estos corresponden a los casos de redirecciones en donde el crawler perdió el rastro de la URL inicial.

Se midió la distribución del porcentaje de descarga de las referencias por elemento para aquellos que tenían al menos una referencia. Este grupo de elementos corresponde a 3.899.953, 71 % del total. A pesar de que solo el 11 % de las referencia fueron descargadas y asociadas a un elemento, se esperaba encontrar una distribución de descargas por elemento relativamente uniformemente. Sin embargo, se vio que solo 1.136.477 elementos tuvieron 1 o más referencias descargadas, lo que corresponde al 29 % del total de elementos que contenían al menos 1

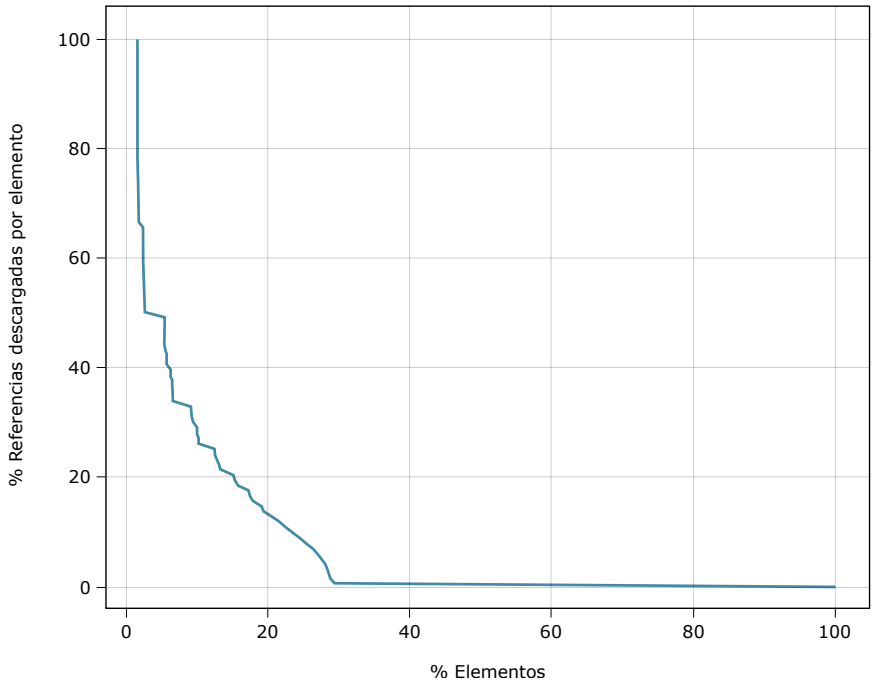

Figura 5.3: Distribución del porcentaje de elementos con respecto al porcentaje de descarga de sus respectivas referencias. Mientras el 100 % de los elementos tiene al menos 0 % de sus referencias descargadas, 29 % tiene al menos 1 % de referencias descargadas y 2 % de elementos tiene el 100 % descargado.

referencia. La figura 5.3 muestra esta distribución dispareja del porcentaje de descargas por elementos. La consecuencia de esto es que al buscar referencias de manera offline, la mayoría de los elementos no contarán con referencias asociadas, por lo que se tendrá que limitar la búsqueda a todo el índice para estos elementos.

También se midió la distribución de descargas esta vez considerando los *pay-level domains*. Se quiso ver si la distribución que realizó el crawler fue pareja por dominios o dio preferencia a algunos por sobre otros. Primero se observaron los top 10 pay-level domains entre el total de URLs y las indexadas, encontrándose que solo 3 de los dominios más populares están en el top 10 de las referencias indexadas, como lo muestra la tabla 5.1. Se graficó el número de documentos descargados para los 10 pay-level domains con más URLs del total de enlaces, encontrándose una distribución dispareja de descargas. Mientras algunos dominios concentran una gran cantidad de descargas, otras presentan casi ninguna, como se ve en la figura 5.4. En general se ve que el crawler prioriza la descarga de dominios tomando en cuenta la relevancia y la cantidad que el dominio representa del total.

5.1.3. Muestra de entidades

La validación del desempeño de la solución se esperaba realizar con el índice completo de documentos teniendo todas las referencias descargadas. Sin embargo, dado el lento progreso, los elementos tuvieron solo una pequeña parte de sus referencias asociadas descargadas. Para

Pos.	Set completo de URLs	URLs Indexadas
1.	archive.org	bbc.co.uk
2.	doi.org	nytimes.com
3.	nih.gov	archive.org
4.	nytimes.com	billboard.com
5.	bbc.co.uk	newspapers.com
6.	webcitation.org	thegazette.co.uk
7.	allmusic.com	sports-reference.com
8.	youtube.com	reuters.com
9.	theguardian.com	baseball-reference.com
10.	archive.is	bbc.com

Tabla 5.1: pay-level domains con más URLs para el total set de referencias y las indexadas del mayor al menor

no medir la solución con elementos con referencias "incompletas", se generaron muestras de entidades para las cuales se les descargaron todos sus documentos.

Se tomaron 5 grupos de 1000 elementos cada uno elegidos al azar dentro de un determinado rango de identificadores, para hacer una evaluación de elementos con distintos grados de popularidad y número de referencias en Wikipedia. Debido a que Wikidata asigna incrementalmente identificadores a los elementos, en general los más populares y con más referencias tienen identificadores menores. Se eligieron grupos en rangos de identificadores progresivamente mayores, por lo que el primer grupo contiene en general los elementos más populares y el último grupo los menos populares. Los rangos de identificadores se distribuyeron de la siguiente forma:

- **Grupo A**: Q1 - Q10.000
- **Grupo B**: Q10.001 - Q100.000
- **Grupo C**: Q100.001 - Q1.000.000
- **Grupo D**: Q1.000.001 - Q10.000.000
- **Grupo E**: Q10.000.001 - Q65.000.000

Para estos elementos se descargaron las referencias de Wikipedia y se añadieron a un índice aparte. En la tabla 5.2 se resume la cantidad de enlaces obtenidos (contando repeticiones).

En promedio se indexaron solo el 54 % del número total de enlaces de los artículos de Wikipedia. Hay distintas razones de por qué esto sucede: por una parte el crawler filtra varias URL como aquellas que tengan una consulta en el query string o estén mal formadas; otras entregan errores 4xx o 5xx al intentar descargarlas o son descartadas internamente debido a su contenido, etc. Como era esperable el grupo A contiene la mayor cantidad de

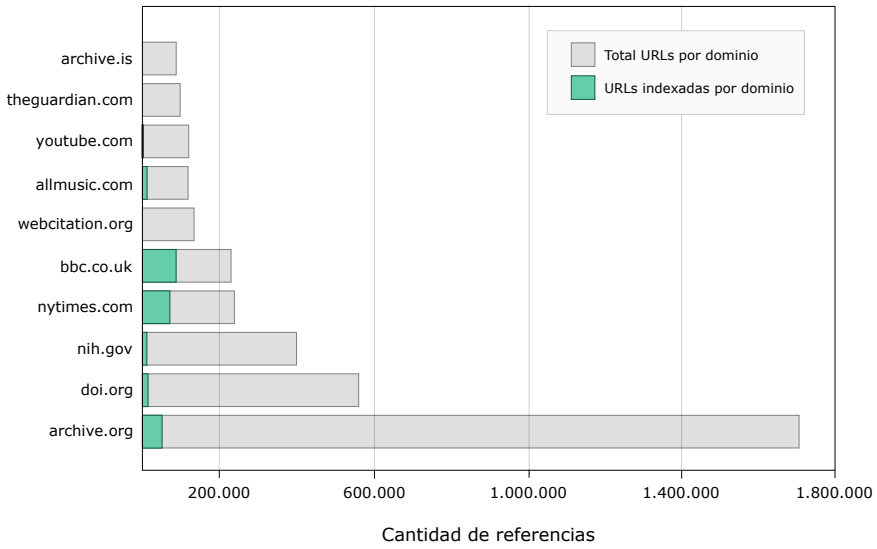

Figura 5.4: Cantidad de documentos descargados para los 10 pay-level domains con más enlaces del total de URLs. Datos se encuentran en apéndice A.1.

	Grupos					
	A	B	C	D	E	Todos
Total URLs de Wikipedia	40.666	12.763	7.111	4.917	5.365	70.822
URLs indexadas	22.268	6.945	3.682	2.399	2.945	38.239

Tabla 5.2: Cantidad de documentos total e indexados para los distintos grupos

referencias, y a partir de ahí comienzan a disminuir. Sin embargo el grupo E contiene en total más referencias que el grupo D, indicando que a partir de esos rangos de identificadores el número de referencias por elemento ya no depende del identificador.

Los dos principales tipos de validaciones que se realizaron fueron: usar las actuales referencias de Wikidata y mediante la construcción de *gold standards*. Para ambos se hicieron mediciones de rendimiento comparando los resultados de la solución con una línea base de resultados aleatorios.

5.2. Referencias conocidas de Wikidata

La primera prueba fue verificar el rendimiento del sistema usando las referencias conocidas de Wikidata; se midió si el sistema lograba retornar estas mismas al buscar referencias. Para encontrar las URLs en común se compararon, por cada elemento y sin hacer distinción por grupo, las referencias indexadas con las oficiales usadas en los hechos del respectivo elemento

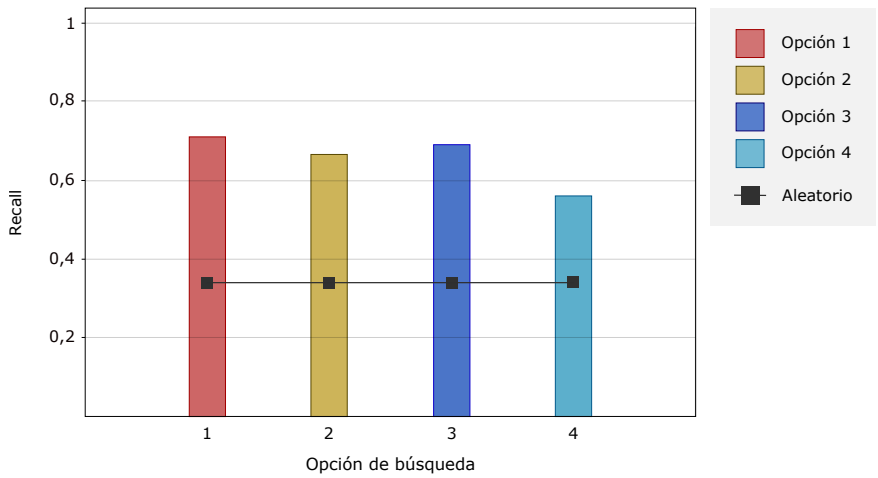

Figura 5.5: Comparación de recall para las opciones de búsqueda usando las referencias existentes conocidas de Wikidata. Datos se encuentran en apéndice A.2.

en Wikidata. Lo primero que se encontró fue un bajo porcentaje de reutilización de enlaces de Wikipedia en hechos de Wikidata, solo 74 elementos de los 5000 (1,4 %) tenían alguna referencia que era igual a una de Wikipedia y del total de 38.239 URLs indexadas solo 163 (0,42 %) eran usadas en algún hecho de Wikidata para validarlo (incluyendo repeticiones). Aunque hay que mencionar que para encontrar referencias iguales se hizo una comparación exacta de URLs, por lo que páginas iguales con distintas URLs fueron consideradas distintas. Esto en parte puede explicar el bajo número de referencias en común en Wikipedia y Wikidata.

Usando estas 163 URLs se realizó una prueba de *recall* por cada una de ellas probando las opciones de búsqueda mencionadas en la sección 4.5. Se consideró el top 3 de los documentos retornados y el rango de búsqueda a aquellas referencias correspondientes a cada elemento. El promedio de los resultados y línea base aleatoria se muestran en la figura 5.5.

La solución entregó en promedio por opción de búsqueda un valor de 0,66 para el recall, mientras que la línea base aleatoria 0,37. En este caso la solución demostró buenos resultados en general para todas las opciones de búsqueda; aunque hay que considerar que el 99 % de las afirmaciones contenía solo una referencia que el sistema debía retornar.

5.3. Gold Standards

La segunda prueba que se realizó se basó en crear gold standards para 5 elementos de cada grupo de la sección 5.1.3, en total 25 elementos. El gold standard de cada elemento consistió en registrar manualmente para cada afirmación aquellas páginas indexadas (de su página de Wikipedia) que la validaban. Los 5 elementos de cada grupo para este fin no fueron elegidos al azar, sino que, para evitar la inclusión de algunos con muchísimas referencias o pocas referencias, se tomaron de acuerdo al número de referencias asociadas de su artículo de Wikipedia que se tenían indexadas, de tal manera que fuera una cantidad cercana al

promedio de ese grupo.

La cantidad de afirmaciones y referencias en promedio de cada grupo se muestran en la tabla 5.3. Se puede ver que a medida que los identificadores aumentan por grupo, el número promedio de afirmaciones y referencias indexadas disminuye.

	Grupos					
	A	B	C	D	E	Todos
Promedio afirmaciones por elemento	48	18	17	13	8	21
Promedio URLs por elemento	23	7	4	2	3	7

Tabla 5.3: Cantidad promedio de afirmaciones y URLs de los distintos elementos al usar gold standards por grupo.

Para evaluar el rendimiento de la aplicación se usaron solo aquellos hechos que eran validados por al menos un documento, las cuales en promedio correspondieron al 37 % del total de afirmaciones de todos los grupos. El detalle de la cantidad de hechos validados por grupo se muestra la tabla 5.4.

	Grupos					
	A	B	C	D	E	Todos
Hechos validados por al menos 1 URL	42 %	27 %	26 %	52 %	31 %	37 %

Tabla 5.4: Porcentaje de afirmaciones validadas por grupo.

Con este grupo de hechos se probaron las 4 opciones de búsqueda de la sección 4.5 usando 3 tipos de pruebas distintas: nDCG, any at k: 1-5, y F_1 score.

5.3.1. nDCG

Dado que DCG realiza una medición de relevancia dados los resultados de una consulta, esta prueba nDCG se hizo para tener una idea general del rendimiento de la solución, tomando en cuenta todos los resultados retornados por afirmación. Se usó la versión normalizada de DCG dado que los todos elementos tenían una cantidad variable de documentos indexados. Concretamente, se le asignó un puntaje de 1 a los documentos que aparecían en el gold standard y 0 a los que no, sin hacer distinción por relevancia (es decir, para todas las consultas a los documentos se les asumió la misma relevancia). Se realizó la prueba para cada grupo de elementos y para cada método de búsqueda. Los resultados se muestran en la figura 5.6. Es esperable encontrar mayores puntajes entre los últimos grupos ya que en general estos tienen

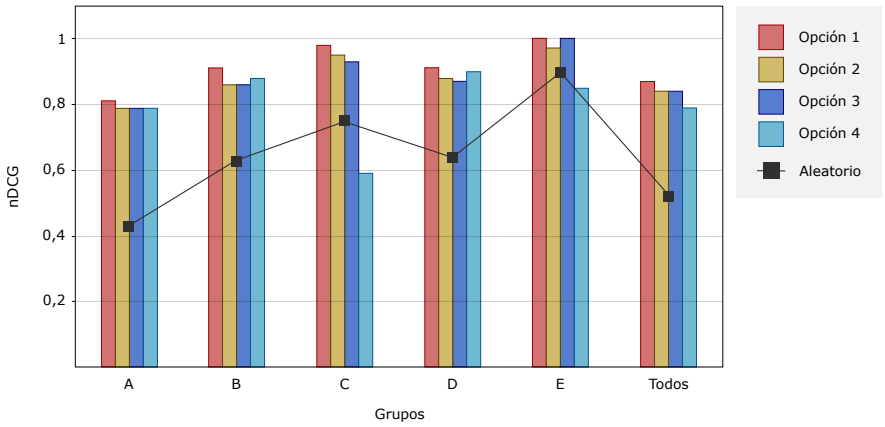

Figura 5.6: Comparación de las distintas opciones de búsqueda y línea base aleatoria por grupos mediante la medición nDCG. Datos se encuentran en apéndice A.3.

una menor cantidad de referencias totales por elemento. En este caso la opción de búsqueda 1 consiguió resultados levemente mejores que el resto.

5.3.2. Any at k: 1-5

Dado que en general el usuario va a querer encontrar solo *una* referencia válida, esta prueba consistió en medir si el programa retornaba al menos 1 referencia para cada hecho en los primeros k resultados, para k = [1, 2, 3, 4, 5]. En este caso, dada la variable adicional de k, no se separaron los resultados por grupo. Los resultados se muestran en la figura 5.7. La idea principal de medir esto es ver la cantidad óptima de resultados o documentos a mostrar al usuario, ya que lo ideal es mostrar la menor cantidad posible pero una cantidad razonable ya que no siempre el primer resultado va a ser uno que valide la afirmación. En general para todos los métodos se ven los mayores aumentos en los resultados al llegar a k = 3 y a partir de allí empieza a aumentar más lentamente la efectividad, por lo que 3 resultados se eligió como la cantidad de resultados a mostrar en la aplicación.

5.3.3. F_1

La última prueba usando gold standards que se hizo fue un F_1 score en donde se registró el precision, recall y F_1 para los 3 primeros documentos retornados, ya que la solución muestra 3 referencias por hecho. Los resultados se muestran en la figura 5.8. Debido a la baja cantidad de afirmaciones y referencias disponibles para los dos últimos grupos, no se ven diferencias de rendimiento entre la solución y la línea base aleatoria. Sin embargo, para los primero grupos sí existe una diferencia notoria visible, especialmente para el grupo A que muestra el mejor rendimiento. En general entre mayor es la cantidad de referencias disponibles por elemento, mejores resultados tiene la solución por sobre la línea base.

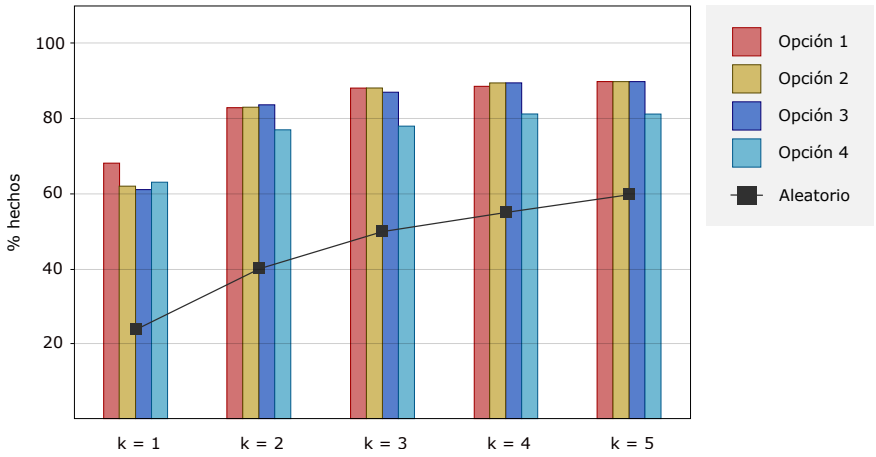

Figura 5.7: Porcentaje de afirmaciones de los distintos grupos al medir si lograban retornar al menos 1 referencia válida en los primeros k resultados, comparando las opciones de búsqueda. Datos se encuentran en apéndice A.4.

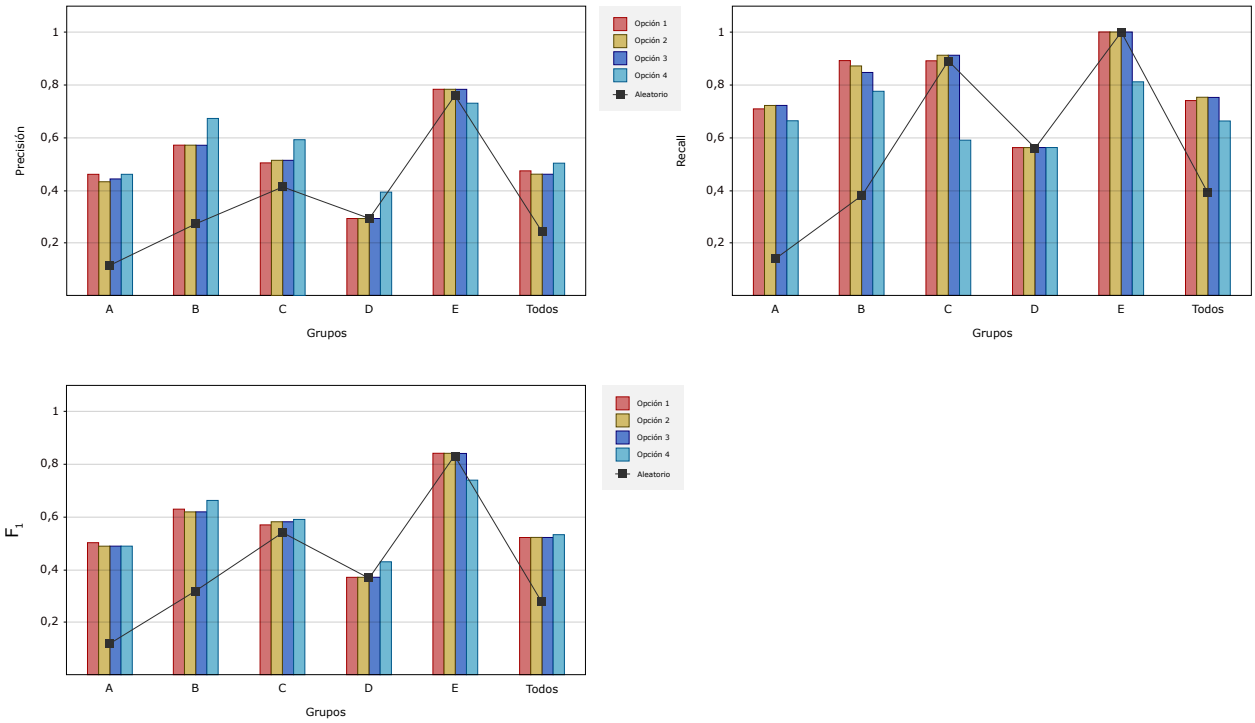

Figura 5.8: Resultados del rendimiento de la solución mediante F_1, comparando las distintas opciones de búsqueda y resultados aleatorios por por grupo. Datos se encuentran en apéndice A.5.

5.4. Validación de extractos

Se probó el rendimiento de los extractos de texto del contenido de la página que entrega la aplicación usando los mismos 25 elementos de las pruebas anteriores. Del total de 522 afirmaciones, se tomaron aquellas que eran validadas por alguna referencia en los 3 primeros resultados. Se midió, para las opciones de búsqueda 1 y 3, el porcentaje de extractos que efectivamente servían para identificar si la página validaba el hecho. Cada opción contempló 171 hechos para los cuales lograban retornan correctamente referencias. Se encontró que los extractos servían para validar los hechos en el 21 % de los casos para la opción 1 y el 14 % de los casos para la opción 3. La diferencia entre los resultados tiene que ver con que la primera opción mencionada usa menos términos, por lo que el pasaje retornado es más focalizado al hecho. De igual manera ambos porcentajes son bajos considerando que solo se probaron con páginas que se sabía de antemano que validaban el respectivo hecho. Por lo que no se podrá confiar en los extractos para para identificar si los documentos retornados validan las afirmaciones. Se tendrá que visitar las páginas en la gran mayoría de los casos, precisamente lo que se esperaba evitar. A pesar de estos resultados igual es valioso tener los extractos en la solución, ya que junto con la URL, pueden ayudar a entregar información contextual de la página. Por ese motivo se decidió mantenerlos en la solución.

5.5. Rendimiento offline

Esta prueba consistió en evaluar el sistema usando las referencias que se lograron descargar en el tiempo que se ejecutó el crawl. Para esto se tomaron 50 afirmaciones de 50 elementos sin referencias en sus páginas de Wikipedia elegidos al azar. Luego se midió el porcentaje de afirmaciones para las cuales el sistema lograba retornar una referencias válida en los primero k resultados, para k = [1, 2, 3, 4, 5]. A diferencia de las pruebas anteriores, esta vez se probaron solo las opciones de búsqueda 1 y 3 de la sección 4.5. La idea de realizar esta medición, además de verificar el funcionamiento offline, es ver cómo afecta el rendimiento del sistema al usar elementos que originalmente no tienen referencias, y ver si es necesario o no que estos tengan referencias asociadas para poder encontrar documentos que validen sus afirmaciones. Los resultados se muestran en la figura 5.9. Hay que tener en cuenta que el índice solo contenía el 14 % de URLs indexadas del total extraído de Wikipedia, por lo que en ese sentido se esperaba encontrar bajos resultados. La opción de búsqueda 1 retornó para el 16 % de las afirmaciones un documento que las validaba en los primero 3 resultados, mientras que la opción 3 solo el 8 %. Para todas las afirmaciones la opción 3 logró encontrar referencias correctas solo en los casos en los que la opción 1 también lo hacía, lo que demuestra que general, dada la gran cantidad de documentos al usar el índice completo, extender los términos de búsqueda no mejora los resultados en este caso.

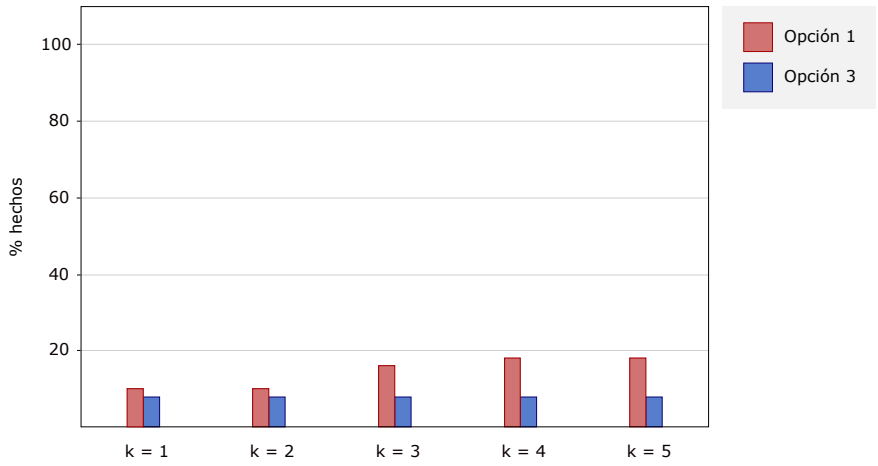

Figura 5.9: Resultados del rendimiento de la solución mediante F_1, comparando las distintas opciones de búsqueda y resultados aleatorios por por grupo. Datos se encuentran en apéndice A.6.

Capítulo 6

Conclusión

Este trabajo se enfocó en ayudar a dar solución al problema de Wikidata sobre las formas que tiene actualmente para buscar referencias. La idea de la solución que se implementó es simple: usar las existentes referencias de Wikipedia para hechos de Wikidata. Este método es el primero que sugiere Wikidata a sus colaboradores, pero la diferencia consistió en que esta herramienta hiciera el trabajo de forma rápida y automática, para evitar que los colaboradores tengan que realizarlo manualmente. Esto se logró mediante la creación de un índice con las páginas de la sección de Referencias de Wikipedia, para luego consultarlo para cada afirmación con el método TF-IDF con el nombre del elemento, propiedad y valor, retornando un conjunto de posibles páginas que sirvan de fuente junto con un pequeño extracto de coincidencias de los términos de búsqueda.

El rendimiento de esta solución es dependiente principalmente de 3 cosas: las referencias de Wikipedia, el funcionamiento de TF-IDF para retornar referencias relevantes y los extractos de las páginas para no tener que visitarlas cada vez. Según las pruebas realizadas con elementos de distintas cantidades de referencias y afirmaciones, se puede decir que en general las referencias de Wikipedia lograron validar una cantidad baja de afirmaciones. Se vio que en promedio el 37 % de las afirmaciones lograron ser validadas por al menos 1 referencia obtenida del artículo de Wikipedia. Sin embargo, TF-IDF obtuvo buenos resultados en general para todas las pruebas que se hicieron (nDCG, F_1) y para todas las opciones de búsqueda en comparación a la línea base de resultados aleatorios. En promedio, el sistema entregó al menos una referencia válida en los primeros 3 resultados en el 88 % de los casos (considerando solo el 37 % de los hechos que eran validados por alguna referencia en Wikipedia). Los extractos de texto así como la URL demostraron ser una ligera ayuda para dar un indicio de qué trata la página a grandes rasgos. Solo una pequeña cantidad de extractos logró mostrar la parte en donde se valida la información, lo cual era esperable dado que en general la afirmación no se valida en una sola frase, y el extracto solo corresponde a una sección de texto continuo.

Al inicio del trabajo se extrajeron alrededor de 23.200.000 referencias de Wikipedia y se esperaba descargar e indexar todas ellas. Se asumió que el tiempo en realizar el crawling de estas páginas fueran un par de meses, pero este tiempo se subestimó y en realidad tomó mucho

más de 6 meses. La causa de esto fue probablemente porque no se configuró óptimamente Nutch™ al momento de iniciar el crawl. Se priorizó empezar la descarga de documentos lo antes posible en vez de analizar las mejores opciones para el crawl. Debido a esto el uso de la herramienta no será el óptimo. En las pruebas realizadas el sistema logró retornar referencias válidas solo para el 16 % de afirmaciones en los primeros 3 resultados al usar todo el índice. Este es un valor muy bajo, ya que solo se cuenta con el 14 % de documentos indexados del total. Si se hubiesen descargado más páginas, se espera que los resultados para el uso de la herramienta fueran mejores. Dada la cantidad parcial de documentos del índice, no se puede hacer una evaluación completa de la herramienta.

El punto más importante para lograr una buena solución tiene que ver con la calidad y cantidad de las referencias, ya que todo el resto se basa en eso. En general para los elementos con los que se probó, considerando los que se les descargaron sus referencias, las páginas de Wikipedia no fueron suficientes para validar la mayoría de los hechos. Lo más importante para mejorar la solución sería extender el set de páginas autoritativas en donde buscar referencias. Por ejemplo, en este trabajo se usaron solo las referencias de la versión en inglés de Wikipedia, por lo que una primera alternativa sería extraer las referencias de los artículos en otros idiomas y ver cómo eso mejora la solución. También se podría relacionar de otra manera las referencias extraídas con los elementos, por ejemplo para el elemento "Santiago" se buscan referencias en el artículo de Wikipedia de Santiago, sin considerar que las referencias de los artículos "Región Metropolitana de Santiago" o "Chile" pueden servir de igual manera. Otro método para obtener más referencias sería realizar un proceso de búsqueda automática directamente en Google, usando los datos de cada afirmación. Esto presentaría otros desafíos como el que se deban retornar solo páginas autoritativas, ya que al buscar páginas en Google no es seguro que se visitarán solo páginas confiables, lo cual es asumido al buscar en las referencias de Wikipedia. En relación a los extractos de texto de las páginas es evidente que la opción que ofrece Solr™ es muy simple para este caso de uso. Se necesita algo más sofisticado que pueda retornar un pedazo más elaborado de texto, que incluya muchas veces distintos pasajes, pero relacionados, de una misma página.

A pesar de que el método TF-IDF mostró buenos resultados, tiene algunas falencias que puede mejorar. Por ejemplo cuando una afirmación es validada dentro de una página implícitamente, un método como TF-IDF que solo busca coincidencias de términos falla. Al querer validar que una persona nació en Chile, puede ocurrir que una página lo valida diciendo que nació en Santiago (y por consiguiente en Chile). Para identificar estos casos como referencias válidas, va a ser necesario usar un método más inteligente que vaya un poco más allá y pueda razonar relaciones de términos.

Esta solución no podrá ser definitiva para el problema de Wikidata de encontrar referencias, ya que los resultados no cumplen con tener un alto grado de efectividad por afirmación. Sin embargo es un aporte y apunta en una buena dirección para encontrar una herramienta efectiva. Una solución ideal no debe contemplar dejar el trabajo de buscar referencias a los colaboradores, sino que debe funcionar de forma automática y delegarles a ellos solo un rol de verificación de la referencias encontradas por el programa. Luego se puede seguir trabajando para que en un futuro la adición de referencias esté completamente automatizada y no requiera el trabajo de colaboradores, para así disminuir la cantidad de hechos sin referencias al mínimo.

Glosario

API *Application Programming Interface* corresponde la interfaz o reglas que define un programa para que otros programas puedan interactuar con él. 27

Endpoint Es un extremo dentro del contexto de un canal de comunicación en la web. 2

GET Una de las acciones definidas en el protocolo HTTP que indica que solo se quiere obtener datos. 15

HMTL *Hypertext Markup Language* es el formato estándar para escribir documentos que son leídos por los navegadores y son mostrados como páginas web. 7

HTTP *Hypertext Transfer Protocol* es el protocolo de comunicación usado para la transferencia de información en la web. HTTP define una serie de verbos al realizar una consulta para indicar la acción que se quiere que se realice. 8

JSON *JavaScript Object Notation* es un formato de archivo para representar información en un formato atributo-valor. Usando frecuentemente para transmitir información en la web. 15

Linked Data Se refiere a la información en la web que se encuentra interconectada y es legible por programas, esto permite que los datos tengan una relación contextual con lo que se puede inferir más información. 2

Pay-level domains También conocidos como dominios de segundo nivel, son los dominios que siguen a los dominios de primer nivel. Por ejemplo, *.cl*, *.com* y *.org* son dominios de primer nivel, mientras que *mineduc.cl*, *google.com* y *wikipedia.org* son dominios de segundo nivel. 41

POST Una de las acciones definidas en el protocolo HTTP para indicar que se están enviando datos. 15

RDF *Resource Description Framework* es una especificación para modelar o describir información en la web. 2

SPARQL *SPARQL Protocol And RDF Query Language* es un lenguaje de programación para hacer consultas a bases de datos que tienen su información en formato RDF. 2

TF-IDF *Term Frequency - Inverse Document Frequency* es un valor numérico estadístico que se le da a las palabras dentro de un conjunto de documentos, para expresar qué tan relevantes son para cada documento. Se usa en algoritmos para búsqueda de documentos. i

URI *Uniform Resource Identifier* es un grupo de caracteres que identifica de forma única a un elemento de la web. 8

XML *Extensible Markup Language* es un formato para escribir archivos o documentos de modo que sea legible por programas y humanos. 15

Bibliografía

[1] Tim Berners-Lee. Linked Data. W3C Design Issues, July 2006. From `https://www.w3.org/DesignIssues/LinkedData.html`.

[2] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

[3] Dan Brickley, R.V. Guha, and Brian McBride. RDF Schema 1.1. W3C Recommendation, February 2014. `https://www.w3.org/TR/rdf-schema/`.

[4] Dan Brickley and Libby Miller. FOAF Vocabulary Specification. FOAF Vocabulary Specification 0.99, January 2014. From `http://xmlns.com/foaf/spec/`.

[5] Richard Cyganiak, David Wood, and Markus Lanthaler. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation, February 2014. `https://www.w3.org/TR/rdf11-concepts/`.

[6] Steve Harris, Andy Seaborne, and Eric Prud'hommeaux. SPARQL 1.1 Query Language. W3C Recommendation, March 2013. `https://www.w3.org/TR/sparql11-query/`.

[7] Tom Heath and Christian Bizer. Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136, 2011.

[8] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph. OWL 2 Web Ontology Language Primer (Second Edition). W3C Recommendation, December 2012. `https://www.w3.org/TR/owl2-primer/`.

[9] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[10] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[11] Stanislav Malyshev, Markus Krötzsch, Larry González, Julius Gonsior, and Adrian Bielefeldt. Getting the most out of wikidata: semantic technology usage in wikipedia's

knowledge graph. In *International Semantic Web Conference*, pages 376–394. Springer, 2018.

[12] Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. From freebase to wikidata: The great migration. In *Proceedings of the 25th international conference on world wide web*, pages 1419–1428, 2016.

[13] Alessandro Piscopo, Lucie-Aimée Kaffee, Chris Phethean, and Elena Simperl. Provenance information in a collaborative knowledge graph: an evaluation of wikidata external references. In *International Semantic Web Conference*, pages 542–558. Springer, 2017.

[14] Alessandro Piscopo, Pavlos Vougiouklis, Lucie-Aimée Kaffee, Christopher Phethean, Jonathon Hare, and Elena Simperl. What do wikidata and wikipedia have in common? an analysis of their use of external references. In *Proceedings of the 13th International Symposium on Open Collaboration*, pages 1–10, 2017.

[15] Juan Ramos et al. Using TF-IDF to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142. Piscataway, NJ, 2003.

[16] Barry Schwartz. Google's freebase to close after migrating to wikidata: Knowledge graph impact? `https://www.seroundtable.com/google-freebase-wikidata-knowledge-graph-19591.html`, 2014.

[17] Amit Singhal. Google's official blog, introducing the knowledge graph: things, not strings. `https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html`, 2012.

[18] Bryan Thompson, Mike Personick, and Martyn Cutcher. The bigdata® rdf graph database. In *Linked Data Management*, pages 221–266. Chapman and Hall/CRC, 2016.

[19] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

Anexo A

Datos usados en los gráficos de la sección 5

A.1. Referencias descargadas en top 10 dominios del total de referencias.

Top 10 dominios	Total URLS	URLs Indexadas
archive.org	1.702.143	54.014
doi.org	554.865	15.940
nih.gov	398.293	12.811
nytimes.com	236.149	72.158
bbc.co.uk	232.508	89.053
webcitation.org	135.548	36
allmusic.com	121.442	9.048
youtube.com	119.629	2.968
theguardian.com	100.341	284
archive.is	94.646	94.646

Tabla A.1: Cantidad de referencias descargadas para los dominios pay-level con más enlaces del total de URLs.

A.2. Resultados referencias existentes

	Opción 1	Opción 2	Opción 3	Opción 4	Línea Base
Recall	0.71	0.66	0.69	0.57	0.37

Tabla A.2: Resultados de recall al usar las referencias conocidas de Wikidata con las opciones de búsqueda.

A.3. Resultados validación nDCG

	A	B	C	D	E	Todos
Opción 1	0.81	0.91	0.98	0.91	1	0.86
Opción 2	0.79	0.86	0.95	0.88	0.97	0.84
Opción 3	0.79	0.86	0.93	0.87	1	0.84
Opción 4	0.79	0.88	0.59	0.90	0.85	0.79
Línea base	0.43	0.63	0.75	0.66	0.90	0.52

Tabla A.3: Resultados de la validación nDCG para los distintos grupos y opciones de búsqueda.

A.4. Resultados validación Any at k

	k = 1	k = 2	k = 3	k = 4	k = 5
Opción 1	68 %	83 %	88 %	89 %	90 %
Opción 2	62 %	83 %	88 %	90 %	90 %
Opción 3	61 %	84 %	87 %	90 %	90 %
Opción 4	63 %	77 %	78 %	81 %	81 %
Línea base	24 %	40 %	50 %	55 %	60 %

Tabla A.4: Resultados de la validación any at k para los distintos grupos y opciones de búsqueda.

A.5. Resultados validación F_1

	A			B			C			D			E			Todos		
Opción	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
1	0.46	0.71	0.50	0.57	0.89	0.63	0.50	0.89	0.57	0.29	0.56	0.37	0.78	1	0.84	0.47	0.74	0.52
2	0.43	0.72	0.50	0.57	0.87	0.62	0.51	0.91	0.58	0.29	0.56	0.37	0.78	1	0.84	0.46	0.75	0.52
3	0.44	0.72	0.49	0.57	0.85	0.62	0.51	0.91	0.58	0.29	0.56	0.37	0.78	1	0.84	0.46	0.75	0.52
4	0.44	0.66	0.49	0.67	0.77	0.66	0.59	0.59	0.59	0.39	0.56	0.43	0.73	0.81	0.74	0.50	0.66	0.53
Base	0.11	0.14	0.12	0.27	0.38	0.32	0.41	0.89	0.54	0.29	0.56	0.37	0.76	1	0.83	0.24	0.39	0.28

Tabla A.5: Resultados de los distintos grupos y opciones de búsqueda al medir precisión, recall, y F_1.

A.6. Resultados rendimiento offline

	k = 1	k = 2	k = 3	k = 4	k = 5
Opción 1	10 %	10 %	16 %	18 %	18 %
Opción 2	8 %	8 %	8 %	8 %	8 %

Tabla A.6: Resultados del rendimiento offline de la aplicación comparando las 2 opciones de búsqueda de la solución mediante la validación “any at k”.