UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA CONCEPCIÓN
Facultad de Ingeniería
Ingeniería Civil Informática

RECUPERACIÓN EN UN REPOSITORIO DE DATOS ABIERTOS DE DATASETS RELEVANTES A UN REQUERIMIENTO DE INFORMACIÓN EXPRESADO EN LENGUAJE NATURAL

WEI CHONG LAI VENEGAS

INFORME DE PROYECTO DE TÍTULO PARA OPTAR AL TÍTULO DE

INGENIERO CIVIL INFORMÁTICO

Profesor Guía

José Ignacio Abreu Salas

Concepción, Septiembre 2017

Resumen

Los repositorios de datos abiertos son grandes bancos de información que se encuentran disponibles a todo el mundo, albergando datos de todos los temas y formatos. Uno solo puede llegar a tener miles de datasets, con los cuales se pueden satisfacer requerimientos de información, estos pueden tardar meses en ser analizados desde que son enviados por el solicitante hasta que son respondidos tanto de manera afirmativa (creando un dataset, o señalando que los datos ya se encuentran) o negativamente.

Es por esto que con el presente proyecto se busca proponer un modelo el cual acelere el proceso del análisis de requerimientos de información, formulando un listado de los posibles datasets del repositorio que sean más afines al tema en cuestión. Para esto se presentaron 5 modelos en los cuales se utilizaron diferentes técnicas del procesamiento del lenguaje natural (*tokenize*, etiquetado, obtener sinónimos entre otras). Además en cada listado se hace la pregunta: ¿Es el primer dataset encontrado el correcto?, o ¿En los 5 primeros? y así sucesivamente.

Como idea principal de este proyecto es encontrar en la primera posición del listado lo cual no ocurre en los primeros modelos presentados, pero esto si ocurre en los modelos posteriores. Si bien los resultados de las métricas que se presentan para cada modelo son aceptables (0%-50% de precisión dependiendo del modelo), estas no son representativas ya que se solo se puede utilizar aproximadamente un 1% de los requerimientos de información presentados en el repositorio de datos abiertos.

Abstract

Open data repositories are large information banks that are available to the world, housing data of all issues and formats. One can only have thousands of datasets, with which information requirements can be met, these can take months to be analyzed since they are sent from the requestor until they are answered in the affirmative (creating a dataset, or pointing out that the data is already found) or negatively.

This is why the present project seeks to propose a model that accelerates the process of analysis of information requirements, formulating a list of possible datasets of the repository that are more related to the subject in question. For this, 5 models in which different techniques of natural language processing (tokenize, labeling, synonyms among others) were used. Also in each list the question is asked: Is the first dataset found the right one ?, or ¿In the first 5? and so on.

As the main idea of this project is to find in the first position of the listing which does not occur in the first models presented, but this does occur in later models. Although the results of the metrics presented for each model are acceptable (0 % - 50 % accuracy depending on the model), they are not representative since only about 1 % of the requirements of information presented in the open data repository.

Agradecimientos

Quiero agradecer principalmente a dos personas que me ayudaron a terminar lo que es el ciclo universitario, mi padre y abuela. El me financió la universidad y ella se dedicó a alimentarme durante el tiempo que estuve en esta época y finalmente a una persona que la verdad no conozco personalmente, pero ayudo en la creación de un dataset con lo necesario para realizar este proyecto David Read. Solo puedo decir una palabra para expresar todo lo que siento. Gracias.

Nomenclaturas

Nomenclaturas	**Explicación**
PLN	Procesamiento del Lenguaje Natural
OD	*Open Data*
RDF	*Resource Description Framework*
QA	*Question Answering*
LD	*Linked Data*
QALD	*Question Answering over Linked Data*
KB	*Knowledge Base*
URL	*Uniform Resource Locator*
NLTK	*Natural Language Tool Kit*
CKAN	*Comprehensive Knowledge Archive Network*
BD	Bases de Datos

Índice

Índice de Figuras

.

Índice de tablas

Capítulo 1

Introducción

En el presente capítulo se describirá el tema del proyecto específicamente en objetivos, justificación y metodología que se empleó.

1.1 Presentación del Tema

En este proyecto se utilizaron diversas técnicas de procesamiento del lenguaje natural con la finalidad de encontrar el dataset más relevante a el requerimiento de información dado, entre todo un repositorio de datos abiertos. Con esto poder resolver de la demora en el análisis de los requerimientos de información,

1.2 Objetivo General

Proponer un método para dada una consulta en lenguaje natural recuperar los datasets más relevantes a ésta, dentro de un repositorio de datos abiertos.

1.3 Objetivo Específico

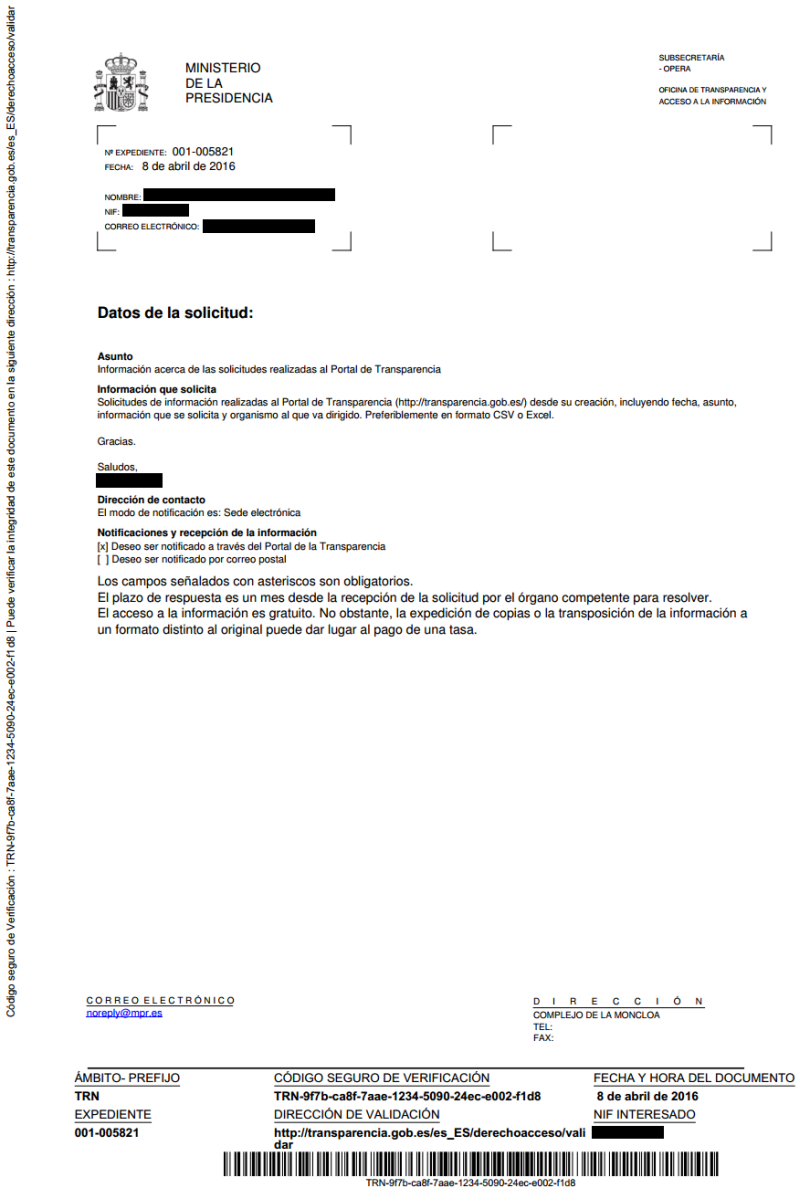
- Estudiar cómo dado un requerimiento de información se recuperan los repositorios relevantes.
- Proponer un enfoque para dado un requerimiento en lenguaje natural, recuperar los repositorios relevantes.
- Validar experimentalmente el enfoque propuesto.

1.4 Justificación

Dada una necesidad de información a ser satisfecha de un repositorio de datos abiertos es responsabilidad de un experto en el tema leer, entender y responder la necesidad de información de este requerimiento, implementando si fuera necesario un nuevo dataset o combinar varios existentes. Este proceso puede llegar a tardar días, semanas e incluso meses. Existen distintos gestores de repositorios de datos abiertos como CKAN o Socrata, además de distintas formas de buscar dentro de estos repositorios de información:

- Consultas directas a los metadatos utilizados en el portal, por ejemplo usando un endpoint (SPARQL) o mediante búsquedas por palabras claves.
- Usando las API expuestas en el portal de datos abiertos.
- Consultas en lenguaje natural que codifican en texto libre el requerimiento de información que tiene el usuario.

Tomando el último ítem como ejemplo, se presenta un requerimiento planteando al portal de datos abiertos de España, el proceso completo desde ingresar el requerimiento a que un especialista tome ese requerimiento, lo interprete y finalmente entregue una respuesta lleva alrededor de 1 mes. La Figura 1.1 muestra un ejemplo de solicitud:



Código seguro de Verificación : TRN-9f7b-ca8f-7aae-1234-5090-24ec-e002-f1d8 | Puede verificar la integridad de este documento en la siguiente dirección : http://transparencia.gob.es/es_ES/derechoacceso/validar

MINISTERIO
DE LA
PRESIDENCIA

SUBSECRETARÍA
- OPERA

OFICINA DE TRANSPARENCIA Y
ACCESO A LA INFORMACIÓN

Nº EXPEDIENTE: 001-005821
FECHA: 8 de abril de 2016

NOMBRE: [redacted]
NIF: [redacted]
CORREO ELECTRÓNICO: [redacted]

Datos de la solicitud:

Asunto
Información acerca de las solicitudes realizadas al Portal de Transparencia

Información que solicita
Solicitudes de información realizadas al Portal de Transparencia (http://transparencia.gob.es/) desde su creación, incluyendo fecha, asunto, información que se solicita y organismo al que va dirigido. Preferiblemente en formato CSV o Excel.

Gracias.

Saludos,
[redacted]

Dirección de contacto
El modo de notificación es: Sede electrónica

Notificaciones y recepción de la información
[x] Deseo ser notificado a través del Portal de la Transparencia
[ ] Deseo ser notificado por correo postal

Los campos señalados con asteriscos son obligatorios.
El plazo de respuesta es un mes desde la recepción de la solicitud por el órgano competente para resolver.
El acceso a la información es gratuito. No obstante, la expedición de copias o la transposición de la información a un formato distinto al original puede dar lugar al pago de una tasa.

CORREO ELECTRÓNICO
noreply@mpr.es

D I R E C C I Ó N
COMPLEJO DE LA MONCLOA
TEL:
FAX:

ÁMBITO- PREFIJO	CÓDIGO SEGURO DE VERIFICACIÓN	FECHA Y HORA DEL DOCUMENTO
TRN	**TRN-9f7b-ca8f-7aae-1234-5090-24ec-e002-f1d8**	**8 de abril de 2016**
EXPEDIENTE	DIRECCIÓN DE VALIDACIÓN	NIF INTERESADO
001-005821	**http://transparencia.gob.es/es_ES/derechoacceso/validar**	[redacted]

TRN-9f7b-ca8f-7aae-1234-5090-24ec-e002-f1d8

Figura 1.1: Solicitud al portal de transparencia de España

Se busca entonces proponer un método para dada una consulta en lenguaje natural recuperar los datasets más relevantes a esta, para ahorrar el tiempo en la etapa del análisis del requerimiento de información.

1.5 Delimitación

Existen una gran cantidad de portales de datos abiertos, pero se opta por el portal de Reino Unido[1], esto porque es uno de los pocos que posee un listado de todos los requerimientos de información hechos a su propio portal.

Además de lo anterior se utilizarán tanto requerimientos de información como repositorios de datos abiertos en el idioma Inglés.

1.6 Metodología

Objetivo: Estudiar cómo dado un requerimiento de información se recuperan los repositorios relevantes

- Se pretende recopilar investigaciones acerca de las distintas técnicas de recuperación de información en repositorio de datos abiertos, con el fin de analizar estos estudios y determinar cuáles técnicas son más relevantes para este proyecto en cuestión.

Objetivo: Proponer un enfoque para dado un requerimiento en lenguaje natural, recuperar los datasets relevantes

- Proponer un flujo de trabajo, el cual permita obtener los dataset más relevantes, utilizando las herramientas más fundamentales para dar cumplimiento a las tareas del flujo.

Objetivo: Validar experimentalmente el enfoque propuesto

- Construir tanto el enfoque propuesto como un corpus de prueba, además de definir las métricas, para validar los resultados obtenidos y finalmente analizar los resultados.

[1] https://data.gov.uk/

La estructura que presenta el proyecto se muestra a continuación:

- Capitulo 1: Introducción
- Capitulo 2: Marco Teórico
- Capitulo 3: Estado del Arte
- Capitulo 4: Preprocesamiento
- Capitulo 5: Descripción de los modelos
- Capitulo 6: Resultados de los experimentos
- Capitulo 7: Conclusión y Trabajo futuro

Capítulo 2

Marco Teórico

Este problema parte de la necesidad de llevar un requerimiento en lenguaje natural a un repositorio de datos abiertos, devolviéndonos un listado de repositorios de datos abiertos congruentes al requerimiento, el proyecto tiene 2 grandes involucrados el procesamiento del lenguaje natural y los datos abiertos. A continuación se presentan algunas de las definiciones que se utilizaran a lo largo del proyecto:

¿Qué es el Procesamiento del Lenguaje Natural?
Es la rama de la ciencia de la computación la cual ve la interacción entre el computador y el humano, específicamente el lenguaje natural humano.

¿Para qué sirve?
Su uso es bastante variado algunos ejemplos de esto puede ser los sistemas QA, la traducción automática, análisis de sentimientos, desambiguadores de palabras, entre otros.

¿Qué es NLTK?

Es una librería *open source* del lenguaje de programación Python el cual se centra en el PLN, esta librería fue creada en el año 2001 en la universidad de Pennsylvania por Steven Bird [1]. Posee 4 principios básicos:

- Simplicidad
- Consistencia
- Extensibilidad
- Modularidad

NLTK posee una vasta literatura como la de Hardeniya [2] y Perkins [3], los cuales en sus documentos explican las funcionalidades de NLTK (Creación de corpus, etiquetados y clasificadores) y sus aplicaciones (Reconocedor de entidades, QA, desambiguador de palabras, clasificador de texto y sistemas de dialogo).

¿Qué es *Open Data*?

Lo que entenderemos por *Open Data* es una práctica en la cual los formatos de datos se dejan en formato libre para todos, en otras palabras sin patentes y restricciones de autor.

¿Para qué sirve?

En especial, estos son utilizados con fines gubernamentales, ya que la gran mayoría de datos del gobierno deben ser públicos, un ejemplo de esto puede ser los datos de la población en ciertos años, estos son almacenados en los llamados repositorios de datos abiertos. Según Bernes-Lee [4] existen 5 niveles de madurez aplicables a los repositorios de datos abiertos, presentados en la Tabla:

Tabla 2.1: *Niveles de Madurez en un repositorio de datos abiertos*

Datos	
☆	Se encuentran disponibles en la web sin importar el formato.
☆☆	Disponibles en formato de datos estructurados, ej. Excel
☆☆☆	Estructurados en formatos no propietarios, ej. CSV
☆☆☆☆	Referencias mediante URIs
☆☆☆☆☆	Contextualizados mediante enlace a otros datos

¿Qué es *Question Answering*?

Es una disciplina de la ciencia de la computación, la cual tiene como objetivo responder preguntas automáticamente expresadas en lenguaje natural no estructurado.

¿Qué es un Corpus?

Corpus o en su plural corpora, se entenderá como "una colección de textos escritos en lenguaje natural para cierto idioma" [5] , que en la actualidad tienen al menos un millón de palabras, se almacena generalmente en un formato electrónico legible por máquinas .

¿Qué es un *Token*?

Los autores Nitin et al. (2010) [6] concluyeron que los *tokens* son elementos de los textos que se identifican en el procesamiento del corpus. Son cadenas de caracteres que se entienden como unidades indivisibles. .

Métricas

A continuación se definen las métricas más comunes en la comparación de algoritmos, y que se utilizaran a lo largo del proyecto:

- *TP: Verdaderos Positivos.*
- *FP: Falsos Positivos.*
- *TN: Verdaderos Negativos.*
- *FN: Falsos Negativos.*

Precision

Total de instancias clasificadas correctamente divididas por el total de instancias [7].

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

Recall

Representa la razón entre el total de instancias verdaderas y total de los documentos que son relevantes [7].

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

F-Measure

Es la combinación de *Precisión* y *Recall* que se utiliza para calcular la puntuación. En el campo de la recuperación de información la *F-Measure* se utiliza con el fin de estimar la clasificación de rendimiento en consultas [8].

$$f - measure = 2 * \frac{precision \ * \ recall}{precision \ + \ recall} \tag{2.3}$$

Capítulo 3

Estado del Arte

Este capítulo se dedica al análisis y estudio de documentos, métodos y métricas que tienen relación con el proyecto. Si bien este tema es relativamente nuevo, se buscan documentos en los cuales existan alguna relación con este proyecto, las investigaciones fueron divididas en las siguientes secciones:

3.1 *Question Answering*

Si bien el tema del proyecto en sí no es sobre de QA, lo que interesa en este tema son los procesos que se utilizaron en estos artículos.

- La universidad Darmstadt [9] propone técnicas para procesar el requerimiento de información, estas se dividieron en 7 etapas consecutivas, partiendo desde un requerimiento de información expresado en lenguaje natural estructurado, realizando *tokenize* a las palabras, clasificando cada una según su correspondencia gramatical (sustantivo, pronombre, entre otros), para luego comprarla con las del corpus. Como conclusión de este trabajo fue que lograron responder un 83% de las consultas presentadas en QALD-1(Serie de competencias en la cual se busca responder la mayor cantidad de querys en el tema de QA), y de este 83% solo el 90% fue respondida correctamente. .

- EL SWNSL (*Semantic Web Search Using Natural Language*) [10] presenta un enfoque de cómo crear una búsqueda en web semántica usando el lenguaje natural, una de los puntos que hay que destacar de este trabajo fue que los requerimientos de información que utilizaron para probar esta herramienta fueron creados por usuarios de Facebook, simplemente explicando que necesitaban en lenguaje natural. Tomaban el requerimiento lo procesaban mediante un NLU (Analizador semántico, etiqueta las palabras y lo deja estructurado para generar la consulta en SPARQL). Teniendo la representación estructurada del requerimiento (*KB Representation*), se lleva a un interpretador semántico el cual genera la consulta.

- ONLI (*Ontology-based Natural Language Interface*) [11] a diferencia de los anteriores utiliza un lenguaje natural estructurado, algunas de las contribuciones más importantes que se deben hacer es el uso de una base de conocimiento de sinónimos para mejorar la clasificación de las palabras. Además a diferencia de SWNSL, este utiliza su propio dominio el cual es DBpedia (Base de datos que ya posee los datos estructurados en el formato RDF), con esto se ve una mejora en el preprocesamiento ya que en este sitio ya están estructurados los datos.

3.2 Acceso en Lenguaje Natural a Base de Datos

- La herramienta creada por Papadakis [12], es capaz de tomar una consulta expresada en lenguaje natural y llevarlo al lenguaje SQL, pero el problema de esta herramienta es que posee una gran cantidad de limitantes, un ejemplo de esto es que las palabras utilizadas en el requerimiento de información deben ser idénticas a las que se encuentran en la base de datos. Lo bueno de esta herramienta es que se puede importar fácilmente un dominio (Archivo o URL) basta con solo cargarlo a la herramienta.

- Ferrandez y Llopis [13] proponen una interfaz expresada en lenguaje natural a bases de datos con accesibilidad a todo usuario, su resultado fue un *benchmark* del 94,8%, su enfoque llamado *AskMe* es completamente portable, no necesitando configuración al cambiarlo de base de datos. Además de poseer su propio ambiente textual de consultas, con lo cual permiten ayudar al usuario en la desambiguación de consultas.

3.3 Acceso en Lenguaje Natural a *Linked Data*

- En este artículo creado por Yahya et al. (2012)[14] se habla de cómo llevar una pregunta expresada en lenguaje natural restringido a una consulta SPARQL, el procedimiento para llevarlo a cabo fue llevar la oración a una estructura de tripletas (entidad, relaciones y clases). Luego de esto resolver las posibles ambigüedades que se presenten, para compararla con la base de conocimiento que en este caso es Yago2.

- SINA [15] es un sistema de búsqueda por palabra clave, el cual toma un requerimiento en lenguaje natural, lo preprocesa, remueve palabras sin influencia del requerimiento (por favor, deseo, quisiera, etc...), determina las raíces de las palabras (Ejemplo casamiento casar), luego de esto genera una sentencia SPARQL, donde la información del dominio se encuentra en *linked data*, más en específico se probó con la base de datos Dbpedia. .

- FreyA [16] es un sistema QA, el cual destaca por ser portable a distintos datasets (Dbpedia, Music Brainz), su principal función es tomar un requerimiento de información expresado en lenguaje natural restringido, estructurarlo y responder el requerimiento. Estas requerimientos se tomaron del QALD-1, si bien sus resultados no son sobresalientes como se dijo anteriormente destaca su portabilidad.

- Este artículo que escribieron Djelloul y Malki [17] contiene un resumen estadístico en el cual se analiza de la gran mayoría de los sistemas QA, entre ellos FreyA [16], siendo este uno de los mejores en rendimiento, se entiende rendimiento para este artículo como:
  - El algoritmo y método de procesamiento de lenguaje natural que uso el sistema.
  - El dominio específico que utiliza el sistema.

3.4 Toma de Requerimientos

- Uno de los trabajos con mayores logros dentro del campo del procesamiento de lenguaje natural fue el que desarrollaron Aithal y Desai [18] , este logra tomar un requerimiento de texto expresado en lenguaje natural, y este llevarlo a tablas, que describen el diagrama de casos de uso del requerimiento de información. Aunque su trabajo es bastante completo este tiene sus limitaciones ya que solo permite requerimiento utilizando voz activa, y además no es capaz de detectar correctamente oraciones compuestas, estas deben separarse en 2 frases , aun con estas limitantes logró generar una representación que es capaz de guiar al Ingeniero de Software a comprender el requerimiento.

- *REVERE* [19] es una herramienta que utiliza el PLN para resumir documentos, el cual puede ser aplicado a cualquier área donde sea necesario resumir una gran cantidad de documentos, principalmente esta herramienta utiliza la búsqueda por palabra clave para generar la síntesis.

3.5 Análisis del estado del arte

En resumen los logros que se ha realizado es tomar un requerimiento de información expresado en lenguaje natural con restricciones, procesarlo detectando las frases con mayor relevancia dentro del requerimiento, para finalmente llevarlo a un modelo estructurado (Diagrama, sentencia, respuesta).

Como conclusión de estos trabajos aún no existe un sistema, herramienta que sea capaz de tomar un requerimiento de información expresado en lenguaje natural sin restricciones, y este entregar un listado de los datasets más relevantes al requerimiento. Algunos de los problemas que se plantean al realizar este proyecto son:

- Agregación: Se entiende como la unión de 2 oraciones, una oración podrá tener un significado, pero al juntarla con otra quizás este significado podría cambiar. Un ejemplo: "Juan compro un chocolate y Juan compro una manzana después".

- Correferencia: Se entiende como la referencia a una entidad entre 2 oraciones, párrafos, etc.... Un ejemplo de esto: "¿Cuál es la capital de Chile?, además de esta obtener los partidos ganados de la selección de fútbol chileno".

- Ambigüedad: Entenderemos esto como la ambigüedad que existen con las palabras (semántica) un ejemplo de esto puede ser:

  "La llama corre por el prado"
  La palabra llama posee 3 significados de llama (llamar), llama (animal) y llama (fuego), puede ocurrir que nuestra oración este escrita sintácticamente correcta, pero aun así esta no tiene coherencia tanto en el dominio como la oración en sí.

Se pretende con este proyecto, dado un requerimiento de información acelerar el proceso de análisis de este, devolviendo los datasets más relevantes al requerimiento dado.

Capítulo 4

Preprocesamiento

Este capítulo describe el proceso de la preparación de los datos para ser utilizados en el proyecto y además todos los pasos a seguir de este. Se presenta una figura en general de lo que sería este proyecto:

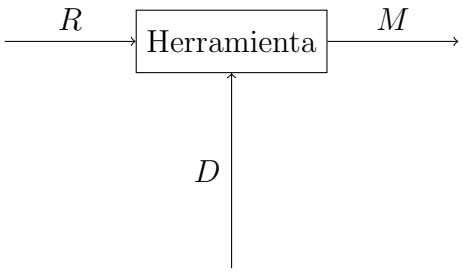


Figura 4.1: Modelo de caja negra del proyecto

R: Requerimientos de información.

D: Datasets del repositorio de información.

M: Modelo de salida.

Las herramientas utilizadas para la implementación de este son:

- Python 2.7
- NLTK 3

4.1 Selección *set* de datos

Para esta etapa se pretende seleccionar los datos tanto de requerimientos de información como de datasets y en que formato se encuentran.

Requerimiento de Información

Entenderemos como un requerimiento de información, una necesidad de información que tendrá un usuario del portal de datos abiertos del Reino Unido. Para este proyecto se utilizaron los requerimientos del portal, los cuales poseen los siguientes campos:

Tabla 4.1: *Muestra Campos Requerimientos de Información*

Numero	Campo
1	Nid
2	URL
3	submitted_by
4	created
5	updated
6	data_themes

El resto de campos se pueden ver en el anexo A.

En total se poseen 803 requerimientos de información, cada requerimiento tiene 14 campos de información. Si bien existen una gran cantidad de requerimientos en el portal, estos no están con todos sus campos completos. A continuación se presenta un gráfico de la cantidad de requisitos vs la completitud de los campos de requisitos:

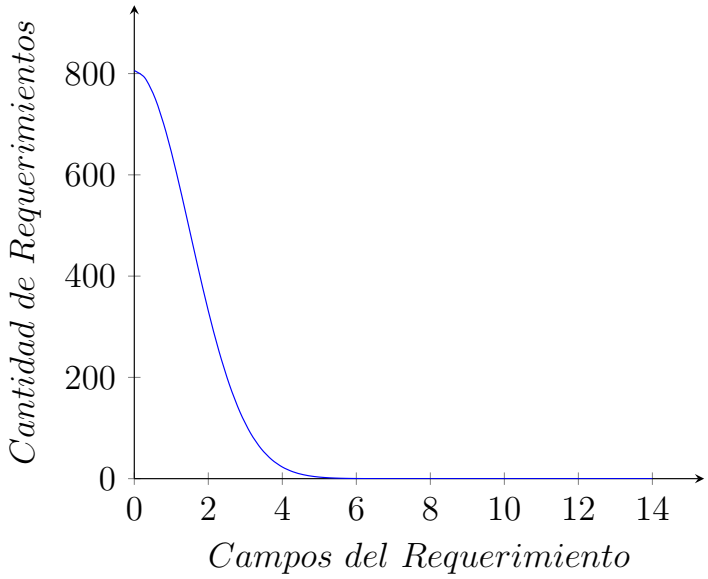


Figura 4.2: Cantidad de Requisitos vs Campos de los Requisitos

Como se observa en el gráfico la cantidad de requerimientos que poseen vs la completitud de los campos de los requerimientos es prácticamente nula cuando se trata de uno que posea todos sus campos. Si bien esta cantidad es bastante baja, es un reflejo de la realidad de muchos portales de datos abiertos en los cuales se cumplen estándares, donde cada requerimiento posea todos los campos pero no necesariamente estos estén completos. Esto puede afectar en los resultados del proyecto de manera tal que cualquiera sea el resultado este no será representativo del propio repositorio.

Dataset

En el portal de datos abiertos del Reino Unido existen un total de 40.674 datasets, los cuales poseen 28 campos algunos de estos son:

Tabla 4.2: *Muestra de Campos Datasets*

Numero	Campo
1	Name
2	Title
3	URL
4	Organization
5	Top level organization

El resto de campos se pueden ver en el anexo B.

Se pretende utilizar los requerimientos de información que poseen un valor en el campo dataset_link el cual sea del portal de datos abiertos del Reino Unido lo que serán 8 datasets.

4.2 Funciones utilizadas en el proyecto

A continuación se presentan las funciones que se utilizaron a lo largo del proyecto

4.2.1 *Tokenize*

Entenderemos como *tokenize* el proceso de separar el texto en partes más pequeñas, ya sean oraciones o palabras.

Oraciones

El texto ingresado se separara en oraciones, usando la función *sentTokenize* [1], la cual se encuentra en NLTK.

$$\underbrace{\textit{I need the average of students in the UCSC in } 2016.\textit{ And I need the number of carrers in the university.}}_{Texto}$$

$$\underbrace{\textit{I need the average of students in the UCSC in } 2016.}_{Oracion1}$$

$$\underbrace{\textit{And I need the number of carrers in the university.}}_{Oracion2}$$

Palabras

Una vez que el texto es separado en oraciones este es separado en palabras, usando la funcion *wordTokenize* [2], presente en NLTK.

$$\underbrace{\textit{I need the average of students in the UCSC in } 2016.}_{Oracion1}$$

$$\underbrace{\textit{I}}_{Palabra}\ \underbrace{\textit{need}}_{Palabra}\ \underbrace{\textit{the}}_{Palabra}\ \underbrace{\textit{average}}_{Palabra}\ \underbrace{\textit{of}}_{Palabra}\ \underbrace{\textit{students}}_{Palabra}\ \underbrace{\textit{in}}_{Palabra}\ \underbrace{\textit{the}}_{Palabra}\ \underbrace{\textit{UCSC}}_{Palabra}\ \underbrace{\textit{in}}_{Palabra}\ \underbrace{2016.}_{Palabra}$$

[1] http://www.nltk.org/modules/nltk.tokenize.html

[2] http://www.nltk.org/api/nltk.tokenize.html

4.2.2 Eliminación de *stopwords*

Se procede a eliminar los *stopwords*[3], entenderemos como *stopwords* a aquellas palabras vacías que no poseen un aporte gramatical a las oraciones.

Un listado de los *stopwords* en el idioma inglés, incluidos en NLTK son:

'i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', 'her', 'hers', 'herself', 'it', 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', 'should', 'now'

[3]http://www.nltk.org/book/ch02.html

4.2.3 Etiquetado

Se procede a realizar un etiquetado, se entenderá esto como el proceso de agregar la etiqueta correspondiente a cada a palabra, estas etiquetas son las distintas categorías gramaticales que existen (adjetivos, artículos, verbos, entre otras categorías), el algoritmo asignará la etiqueta que posea una mayor un sentido para la oración en sí, la función utilizada será *pos_tag*[4] que se encuentra en NLTK. Un ejemplo fue:

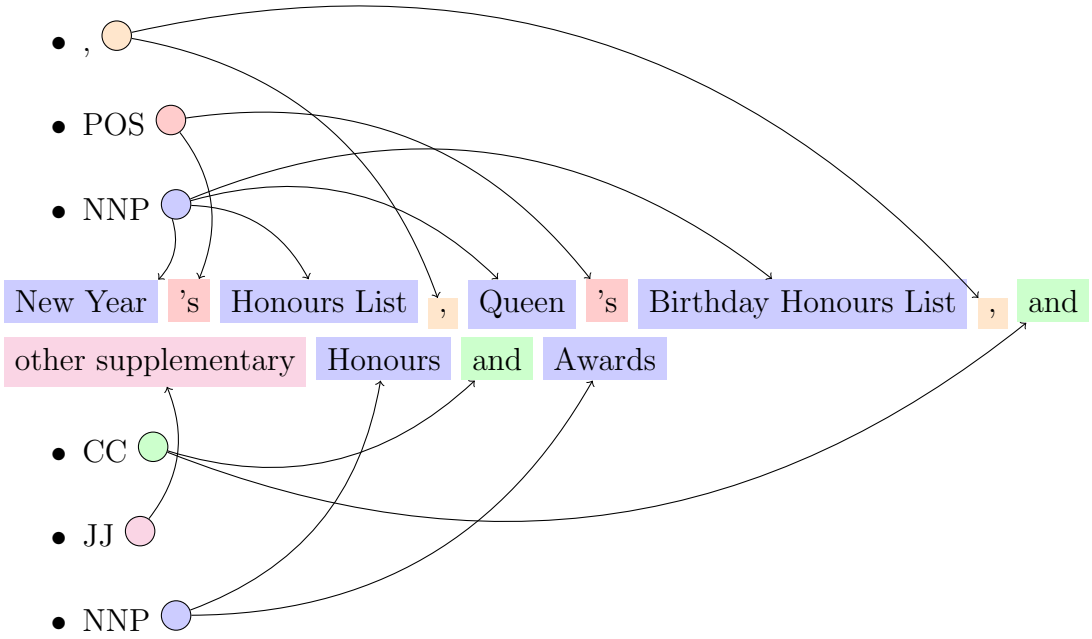


Figura 4.3: Diagrama explicativo del Etiquetado Gramatical

[4]http://www.nltk.org/book/ch05.html

Además a continuación se presenta un listado de las etiquetas presentes en NLTK y su respectiva explicación de etiquetado gramatical:

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential there
5.	FW	Foreign word

Tabla 4.3: *Explicación etiquetado gramatical*

Para ver el resto de etiquetado, revisar el anexo C.

4.2.4 Eliminación de palabras repetidas

En este proceso, se procede a eliminar las palabras repetidas dentro de un texto dado, tomando como ejemplo el requerimiento de información *Honour list* del portal de datos abiertos de UK se tiene:

New Year's Honours List, Queen's Birthday Honours List, and other supplementary Honours and Awards.

Aplicando la función de eliminar las palabras repetidas *set*[5], la cual se encuentra en NLTK, se obtiene:

New Year's Honours List, Queen's Birthday and other supplementary and Awards.

[5]https://docs.python.org/2/library/sets.html

4.2.5 Eliminar caracteres del texto mal identificados

Al realizar el *tokenize* de las palabras existen algunas que no se clasifican correctamente, esto puede deberse a que este mal escrito o NLTK no es capaz de detectarlas correctamente un ejemplo de esto puede ser:

- I'm : Detectara I'm como 3 tokens diferentes (I, ', m)
- words: Esto porque en el requerimiento de información se encuentra escrito de esta manera (word)
- btw: Sigla que en ingles significa: *by the way.*

Para solucionar este problema se utilizará la función *strip* [6], la cual eliminara la palabra o letra que genera problema ('m, :,'s).

$$I'm\ Peter\ from\ Chile \longrightarrow I\ Peter\ from\ Chile. \tag{4.1}$$

4.2.6 Obtener raíz de la palabra

Dada una palabra, se pretende obtener su raíz utilizando la función *lemmatize*[7] que se encuentra en el paquete de NLTK.

$$cars \longrightarrow car \tag{4.2}$$

$$feet \longrightarrow foot \tag{4.3}$$

[6]https://docs.python.org/2/library/string.html

[7]http://www.nltk.org/modules/nltk/stem/wordnet.html

4.2.7 Desambiguar

Existen distintos significados para las palabras dependiendo como se usen, es por esto que uno de los problemas que se presenta en el proyecto es el de la ambigüedad, este se solucionó usando la función *lesk* [8] implementada en NLTK la cual retornara el significado más probable entre el texto en contexto y los *synsets* de WordNet implementados en NLTK.

Ejemplo:

I deposit my money in the bank.

Synsets de palabra bank:

- Synset('bank.n.01'): Lugar donde se deja el dinero
- Synset('bank.n.02'): Conjunto de peces

Synsets resultante:

- Synset('bank.n.01'): Lugar donde se deja el dinero

4.2.8 Obtener sinónimos de la palabra

Una vez desambiguada la palabra, se procede a obtener el sinónimo correspondiente al significado, para expandir el requerimiento. Para esto se selecciona el *synset* [9] correspondiente al mejor significado de la palabra, y obtenemos sus sinónimos.

Money: coin, gold, loot.

[8] http://www.nltk.org/howto/wsd.html

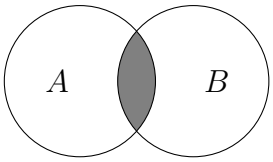
[9] http://www.nltk.org/howto/wordnet.html

Capítulo 5

Descripción de los modelos

En este capítulo se verá en forma general el enfoque que posee cada modelo realizado a lo largo del proyecto.

Una vez que se realizan los pasos mencionados en el Capítulo 4, se procede a hacer un *matching* entre el texto ingresado y el listado de solicitudes del Reino Unido, este *matching* será de manera booleana, es decir, obteniendo las palabras que se encuentran tanto en el requerimiento de información como en el dataset. La Fig. 5.1 muestra una descripción gráfica del modelo booleano [20].



*A:*Requerimiento de Información

*B:*Dataset

Figura 5.1: Figura Matching Booleano

5.1 Modelo 00: Modelo Booleano

En este modelo se realiza un *matching* entre el requerimiento de información ingresado vs la descripción del dataset (Campo número 19 del Anexo B). Se contará las veces que se encuentra la misma palabra tanto en el requerimiento de información como en la descripción del dataset, es decir, si la palabra *hello* aparece 3 veces en el dataset, esta será contada 3 veces, finalmente se verá qué dataset posee una mayor cantidad de palabras repetidas y ese será elegido.

A continuación se presentara el flujo de trabajo del modelo correspondiente:

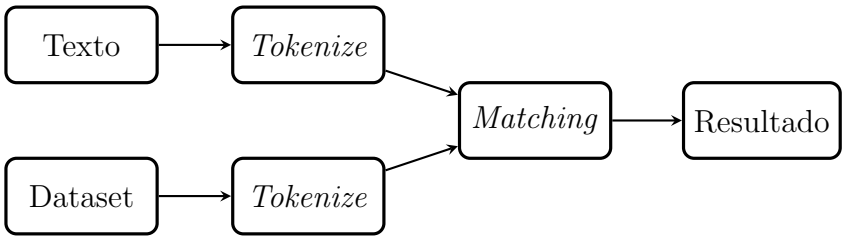


Figura 5.2: Flujo de Trabajo Modelo 00

5.2 Modelo 01: Modelo Booleano eliminando *stopwords*

En este experimento se continua con el *matching* expresado en el modelo 00, contar las palabras repetidas, pero con la diferencia que se eliminaran las *stopwords*, entenderemos como *stopwords* aquellas palabras que no poseen un valor para el análisis del texto, algunos ejemplos de estas palabras pueden ser: 'I', 'Me', 'You, 'Yours', entre otras.

A continuación se presenta el flujo de trabajo correspondiente al modelo:

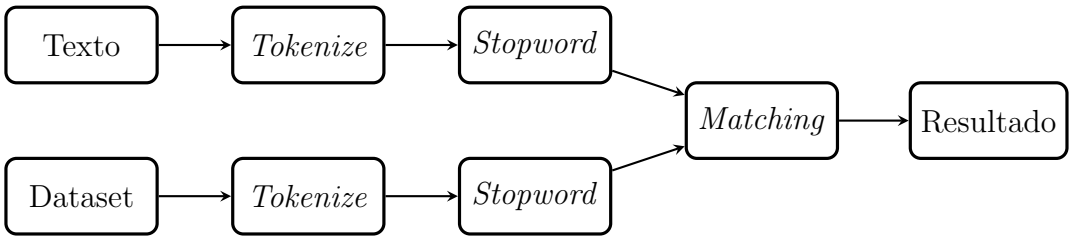


Figura 5.3: Flujo de Trabajo Modelo 01

5.3 Modelo 02: Modelo Booleano eliminando *stopwords* y palabras repetidas

En este experimento se cambia el enfoque del *matching* de los 2 modelos anteriores, se contará la cantidad de palabras diferentes del requerimiento de información que se encuentren textuales en la descripción del dataset, además manteniendo el paso de eliminar los *stopwords* del modelo 01.

A continuación se presenta el flujo de trabajo:

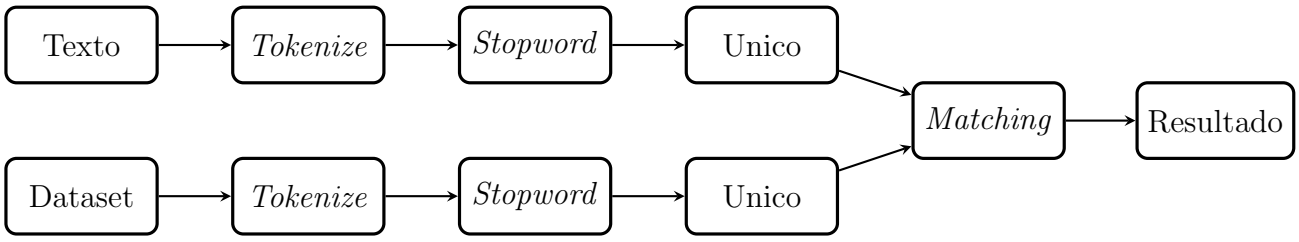


Figura 5.4: Flujo de Trabajo Modelo 02

En la figura 5.4 se agrega un paso más al flujo de trabajo el cual es la eliminación de palabras repetidas, lo entenderemos como *Único*.

5.4 Modelo 03: Modelo booleano *strip*, *lemmatize* eliminando *stopwords* y palabras repetidas

En este experimento se mantiene el enfoque del modelo 02, contar las palabras diferentes encontradas , agregando 2 funciones más al modelo: *strip* y *lemmatize*. La función *strip* se encarga de eliminar aquellos caracteres que no son detectados correctamente ejemplo: *":", "-"*, entre otros. Luego las palabras tanto del requerimiento como del dataset se le realiza *lemmatize*, es decir, se dejaran en infinitivo utilizando la función *lemmatize*, ya que existían palabras que al ser plural el *matching* directo no era capaz de detectarlas (*fish* es distinto a *fishes*).

A continuación se presenta el flujo de trabajo correspondiente al modelo:

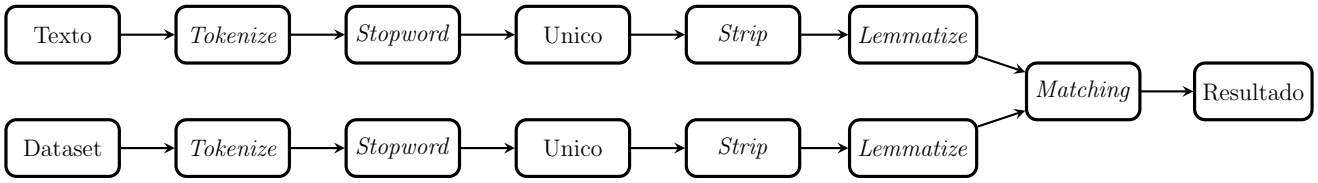


Figura 5.5: Flujo de Trabajo Modelo 03

5.5 Modelo 04: Modelo booleano *strip*, *lemmatize*, desambiguado, expandiendo la consulta con sinónimos eliminando *stopwords* y palabras repetidas, agregando un valor a cada palabra

En este experimento se mantiene el *matching* contando las palabras diferentes encontradas, además se expande el requerimiento de información, utilizando los sinónimos de las palabras. Si bien para cada palabra existen muchos significados, se desambiguó y se seleccionó el significado más adecuado usando la función *lesk* presente en NLTK. Se agrega al requerimiento los sinónimos correspondientes a cada palabra, y a este conjunto se le realiza *matching* contra la descripción del dataset. Además a cada palabra se le asignará un peso según su etiquetado *postag*, es decir, si la palabra fuera un artículo tendrá un cierto valor, mientras que si es determinante tendrá otro valor y finalmente se toma en consideración que el título del dataset es importante, ya que normalmente entrega información acerca de que habla el dataset en sí, es por esto, que si la palabra se encuentra en el título del dataset también tendrá un valor mayor en comparación a las etiquetas *postag*.

A continuación se presenta el flujo de trabajo correspondiente al modelo:

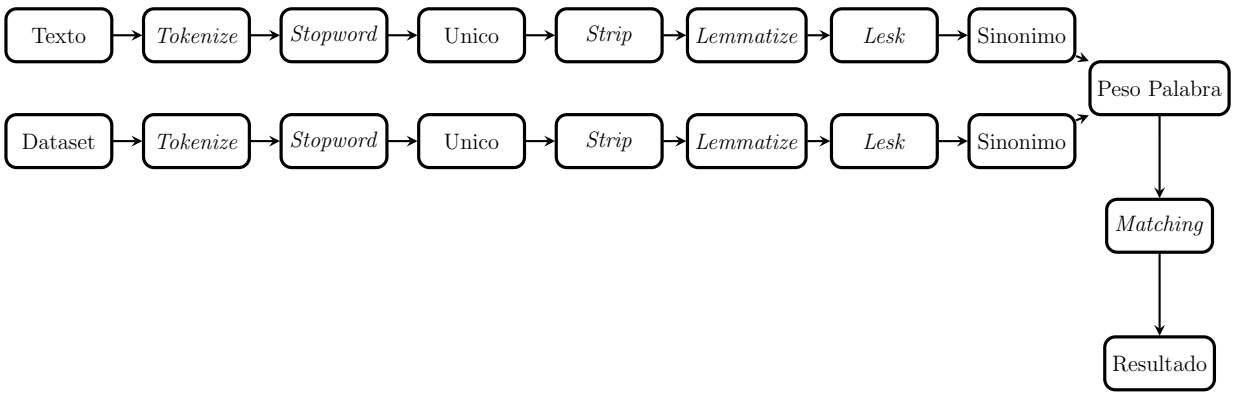


Figura 5.6: Flujo de Trabajo Modelo 04

Capítulo 6

Resultados experimentos

En este capítulo se pretende evaluar la calidad de cada modelo, para esto se utilizaron las métricas mencionadas en el capítulo 2, es por esto que se gráfica los 5 modelos anteriores juntos para observarlos y determinar cual posee un mejor desempeño. La idea de este proyecto es encontrar los dataset más relevantes, para esto se pregunta en primera instancia: ¿El primer dataset es el correcto?, ¿El dataset requerido estará entre los 5 primeros datasets?, ¿O en los 10 primeros?, ¿O los 20?, ¿O 50? y ¿Los 100?

Para los modelos presentados en el capítulo 5 se utilizaran, los siguientes datasets presentes en el repositorio de datos del Reino Unido y su simbología correspondiente a largo del capítulo:

A: Honours lists.

B: Rail Insfractucture.

C: National Noise Model.

D: Road safety from 2014.

E: River network Centrelines.

F: River data.

G: Government expenditure on Bovine Tuberculosis.

H: LSOA Boundaries.

Se utiliza precisamente los 8 datasets, en los cuales se presenta un valor en su campo *dataset_link*. Además se describe la simbología utilizada para las métricas *precision*, *recall* y *F-measure*:

P_n*: Se entenderá como la precisión en el* Top_n

R_n*: Se entenderá como la recall en el* Top_n

F_n*: Se entenderá como la f-measure en el* Top_n

6.1 Resultados modelo 00: Modelo Booleano

A continuación se presenta un ejemplo del modelo propuesto:

Requerimiento de información:

New Year's Honours List, Queen's Birthday Honours List, and other supplementary Honours and Awards.

Descripcion del dataset:

Lists of those who have received honours at New Year and on the Queen's official birthday in June each year since 2012. Information is updated after each List is published and includes the level of award, list, name of the recipient, their occupation / position, and details of citation.

Tabla 6.1: *Cantidad de palabras modelo Booleano*

Palabra	Cantidad de veces repetida la palabra
new	1
year	2
honours	1
list	2
,	5
queen	1
birthday	1
and	3
other	0
supplementary	0
awards	0
.	2

De la tabla 6.1 podemos obtener que el dataset posee en total una relevancia de 18 con respecto al requerimiento de información dado. Se presenta una tabla con los resultados de las métricas:

Tabla 6.2: *Modelo 00*

	P_1	R_1	F_5	P_5	R_5	F_5	P_{10}	R_{10}	F_{10}	P_{20}	R_{20}	F_{20}	P_{100}	R_{100}	F_{100}
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0.1	1	0.18	0.05	1	0.09	0.01	1	0.01
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0	0	0	0.01	1	0.01
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Además se presenta el gráfico de la precisión vs el Top-n:

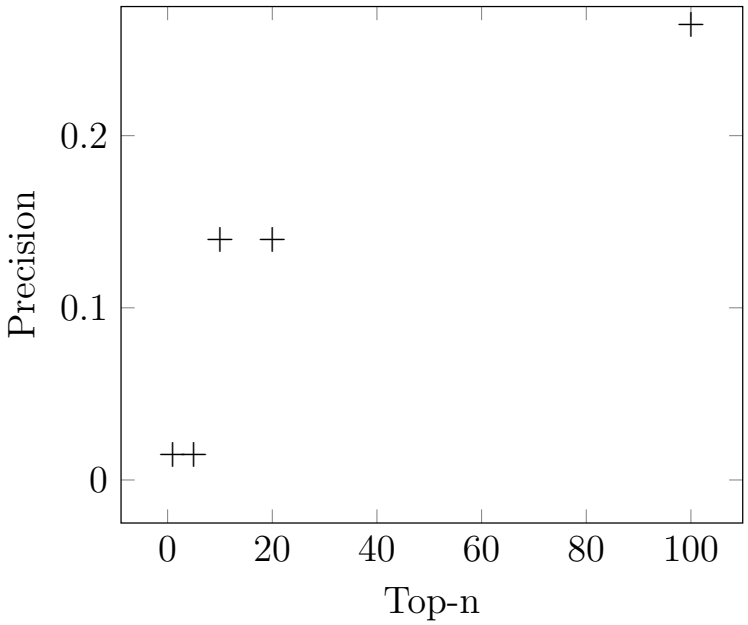


Figura 6.1: Gráfico Modelo 00

Como se observa tanto en el gráfico como en la tabla este modelo no es capaz de encontrar los dataset correctos en los top-1 y top-5, sin embargo es capaz de encontrarlo en el top-10 en adelante. En si este enfoque no es bueno ya que idealmente se espera encontrarlo en el Top-1. Además como se observa en el modelo propuesto este modelo detecta en mayor cantidad los *stopwords* (*","*, *the, and*) en comparación a las palabras que si deberían tener un mayor valor para el modelo.

6.2 Resultados modelo 01: Modelo Booleano eliminando *stopwords*

Se presenta una tabla con los resultados de las métricas:

Tabla 6.3: *Modelo 01*

	P_1	R_1	F_5	P_5	R_5	F_5	P_{10}	R_{10}	F_{10}	P_{20}	R_{20}	F_{20}	P_{100}	R_{100}	F_{100}
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0.1	1	0.18	0.05	1	0.09	0.01	1	0.01
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0	0	0	0.01	1	0.01
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Además se presenta el gráfico de la precisión vs el Top-n:

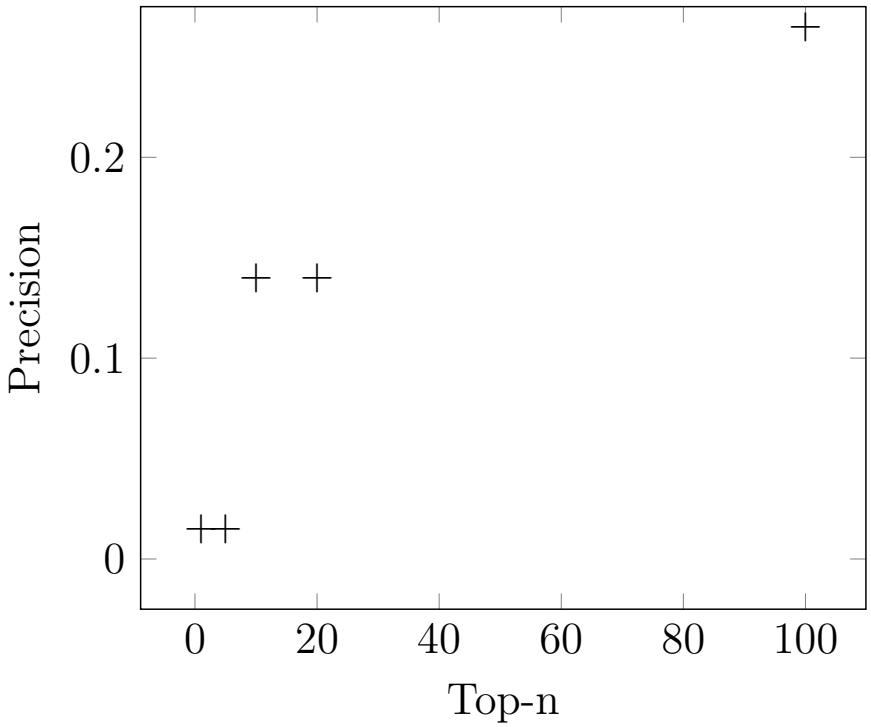


Figura 6.2: Gráfico Modelo 01

Se ve con el resultado que se mantiene prácticamente igual el resultado que con el modelo anterior, podemos sacar como conclusión entre los 2 modelos que a pesar de haber eliminado las *stopwords*, estas no varían el resultado del matching, es decir, las palabras eliminadas (*stopwords*) no variaron este resultado.

6.3 Resultados modelo 02: Modelo Booleano eliminando *stopwords* y palabras repetidas

A continuación se presenta un ejemplo del modelo propuesto:

Requerimiento de información:

New Year's Honours List, Queen's Birthday Honours List, and other supplementary Honours and Awards.

Descripción del dataset:

Lists of those who have received honours at New Year and on the Queen's official birthday in June each year since 2012. Information is updated after each List is published and includes the level of award, list, name of the recipient, their occupation / position, and details of citation.

Tabla 6.4: *Tabla Modelo Booleano eliminando stopwords y palabras repetidas*

Palabras diferentes encontradas
new
year
honours
list
,
queen
birthday
and
other
supplementary
awards
.

De la tabla 6.4 podemos obtener que el dataset posee en total una relevancia de 6 con respecto al requerimiento de información dado. Además flujo de trabajo anterior obtenemos los siguientes resultados:

Tabla 6.5: *Modelo 02*

	P_1	R_1	F_5	P_5	R_5	F_5	P_{10}	R_{10}	F_{10}	P_{20}	R_{20}	F_{20}	P_{100}	R_{100}	F_{100}
A	1	1	1	0.2	1	0.33	0.1	1	0.18	0.05	1	0.09	0.01	1	0.01
B	1	1	1	0.2	1	0.33	0.1	1	0.18	0.05	1	0.09	0.01	1	0.01
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0.2	1	0.33	0.1	1	0.18	0.05	1	0.09	0.01	1	0.01
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0.01	1	0.01
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	1	1	1	0.2	1	0.33	0.1	1	0.18	0.05	1	0.09	0.01	1	0.01

Además se presenta el gráfico de la precisión vs el Top-n:

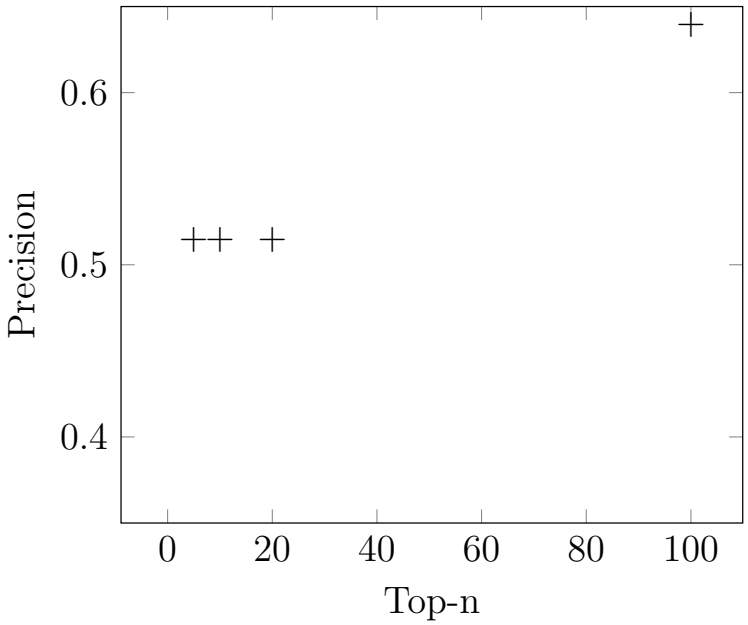


Figura 6.3: Gráfico Modelo 02

Como resultado de esto se ve que el cambio de contar las distintas palabras entre el requerimiento y el dataset en vez de contar las palabras repetidas entre los dos, da un mayor resultado, logrando encontrar resultados en el Top-1. Algunos problemas presentados en los modelos anteriores son:

- Detecta incorrectamente palabras con apostrofes (*I'm* es detectado como 3 caracteres).
- No es capaz de detectar que la palabra *lists* es similar a la palabra *list*, es decir, no es capaz de detectar una similitud en la raíz de la palabra.

6.4 Resultados modelo 03: Modelo booleano *strip*, *lemmatize* eliminando *stopwords* y palabras repetidas

Se presenta una tabla con los resultados de las métricas:

Tabla 6.6: *Modelo 03*

	P_1	R_1	F_5	P_5	R_5	F_5	P_{10}	R_{10}	F_{10}	P_{20}	R_{20}	F_{20}	P_{100}	R_{100}	F_{100}
A	1	1	1	0.2	1	0.33	0.1	1	0.18	0.05	1	0.09	0.01	1	0.01
B	1	1	1	0.2	1	0.33	0.1	1	0.18	0.05	1	0.09	0.01	1	0.01
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0.1	1	0.18	0.05	1	0.09	0.01	1	0.01
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.01
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0.2	1	0.33	0.1	1	0.18	0.05	1	0.09	0.01	1	0.01

Además se presenta el gráfico de la precisión vs el Top-n:

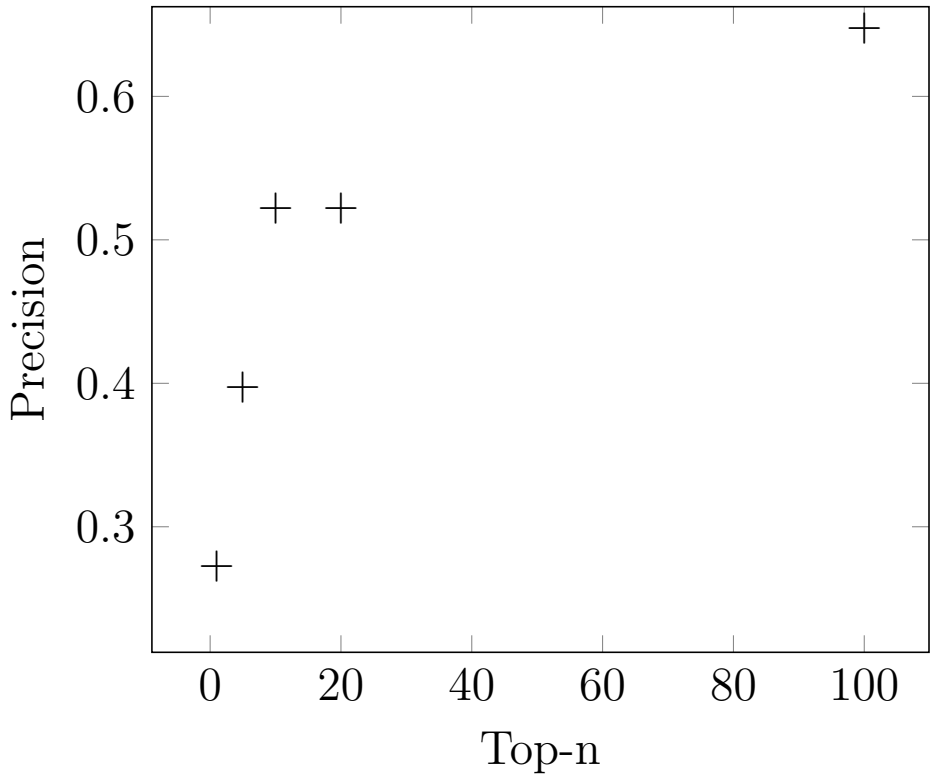


Figura 6.4: Gráfico Modelo 03

Se ve gráficamente que al realizar los 2 procesos antes mencionados, mejora un poco en comparación de los anteriores. Se puede afirmar que los procesos que se agregan en este enfoque mejoran la precisión del modelo.

En este modelo se observa que la precisión en el Top-1 y Top-5 desciende, mientras que en los Top-10, Top-20 y Top-100 se mantiene. Por otro lado como este modelo detecta palabras textualmente similares, no es capaz de detectar palabras que tengan un mismo significado pero sean distintas (sinónimos).

6.5 Resultados modelo 04: Modelo booleano *strip*, *lemmatize*, desambiguado, expandiendo la consulta con sinónimos eliminando*stopwords* y palabras repetidas, agregando un valor a cada palabra

Se presenta una tabla con los resultados de las métricas:

Tabla 6.7: *Modelo 04*

	P_1	R_1	F_5	P_5	R_5	F_5	P_{10}	R_{10}	F_{10}	P_{20}	R_{20}	F_{20}	P_{100}	R_{100}	F_{100}
A	1	1	1	0.2	1	0.33	0.1	1	0.18	0.05	1	0.09	0.01	1	0.01
B	1	1	1	0.2	1	0.33	0.1	1	0.18	0.05	1	0.09	0.01	1	0.01
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	1	1	1	0.2	1	0.33	0.1	1	0.18	0.05	1	0.09	0.01	1	0.01
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	1	1	1	0.2	1	0.33	0.1	1	0.18	0.05	1	0.09	0.01	1	0.01

Además se presenta el gráfico de la precisión vs el Top-n:

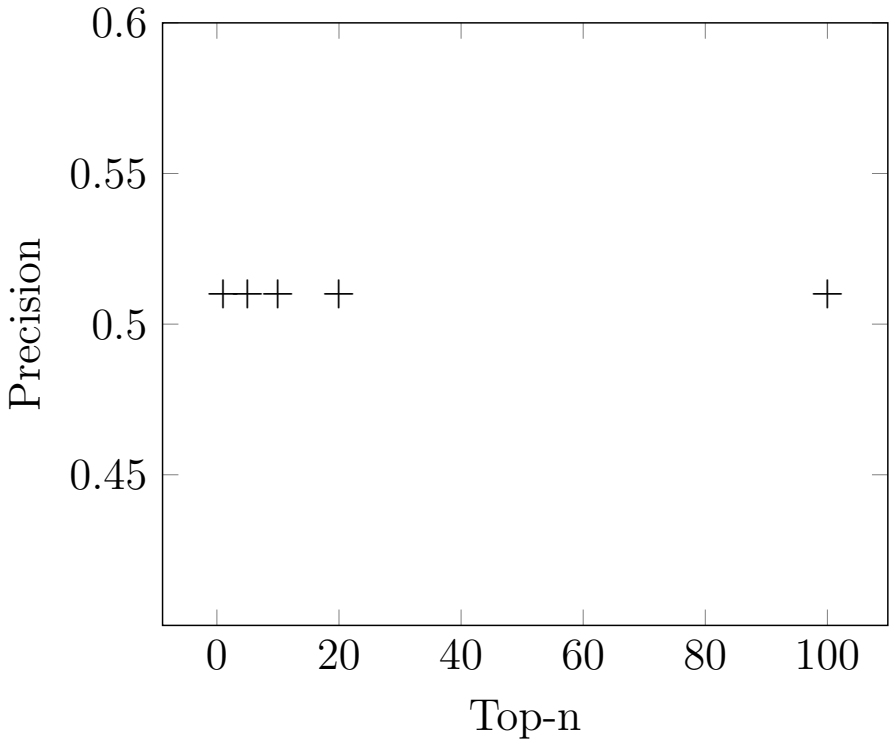


Figura 6.5: Gráfico Modelo 04

Finalmente este último modelo obtiene el mismo resultado sin importar en que Top se encuentre el dataset correcto.

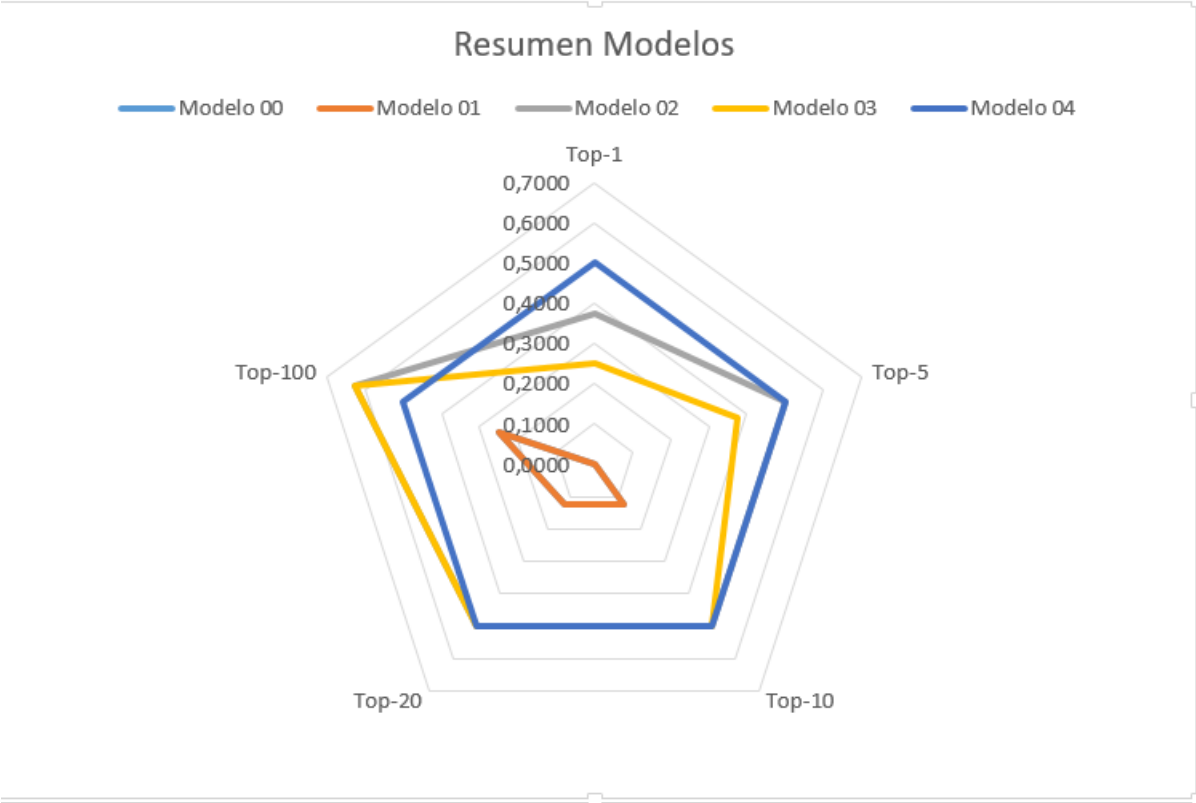


Figura 6.6: Resumen Modelos

La idea general que se plantea en esta herramienta es que, dado un requerimiento de información esta devuelva los datasets más relevantes a este requerimiento.

Según los gráficos 6.1 y 6.2 que corresponden a los modelos 00 Y 01 estos son exactamente idénticos, ¿Porque son idénticos los resultados siendo que se eliminan los *stopwords* en el modelo 01?, una explicación para esto puede ser que los *stopwords* presentes no fueran suficientes como para hacer un cambio significativo en los resultados de los modelos.

Es por esto que se decide cambiar el enfoque en el modelo 02 (contando las palabras distintas que existen), obteniendo mejores resultados en comparación a los dos anteriores. Luego en el modelo 03 se utilizan 2 nuevas funciones: *strip* y *lemmatize* con el fin de obtener un mejor desempeño, lo cual no sucede en primera instancia (Top-1), pero si se mantiene la precisión en los Top siguientes. Finalmente con el modelo 04 se busca expandir la consulta y además darle valor a las palabras que estén más presentes en el requerimiento de información (título del requerimiento), con esto se obtiene un resultado constante en cualquier Top que se encuentre.

Finalmente queda la pregunta: ¿Cual modelo es mejor? ¿Y con cual nos quedamos?, siguiendo la idea original de este proyecto de encontrar el repositorio correcto en la primera posición (Top-1), el correcto sería el modelo 04 (con un total de 4 datasets correctos en la primera posición), sin embargo si analizamos lo que ocurre en el Top-100 del modelo 02 este lo supera ya que encuentra 5 datasets. Es por esto que es importante el análisis tanto del Top-1 hasta el Top-100 ya que el resultado puede cambiar. Resumiendo se decide por el modelo 04 esto porque es la idea del proyecto encontrar el dataset en la primera posición (Figura 6.6).

Capítulo 7

Conclusión y Trabajo futuro

7.1 Conclusión

El principal objetivo de este proyecto fue implementar una herramienta que facilite el proceso de solicitud de datos a portales de datos abiertos. De acuerdo a los objetivos propuestos en el capítulo 1.2, se puede concluir lo siguiente:

Objetivo 1: Estudiar cómo dado un requerimiento de información se recuperan los repositorios relevantes

Si bien este tema como tal es relativamente nuevo, se investigó en distintas fuentes de información reflejadas en las referencias. Se encontraron enfoques en distintas áreas como el acceso a LN a BD, *question answering*, el acceso a LN en *linked data* y la recuperación de información. Sin embargo el tema como tal prácticamente no existe en la literatura, por esto mismo se utilizaron técnicas de otras áreas para aplicarlas en ésta. A través del análisis de los distintos estudios se propuso una serie de pasos a seguir para llegar a un flujo de trabajo, este fue variando según las funciones o procesos que se le agregaron a lo largo del proyecto.

Objetivo 2: Proponer un enfoque para dado un requerimiento en lenguaje natural, recuperar los datasets relevantes

Se preparan flujos de trabajo para cada experimento, luego dado esos flujos se seleccionaron las herramientas o técnicas que fueron necesarias para llevarlos a acabo.

Objetivo 3: Validar experimentalmente el enfoque propuesto

Finalmente se desarrolla la herramienta según cada modelo del proyecto, además se preparó el corpus con requerimientos de información reales, según vimos a lo largo del proyecto este corpus se realiza *tokenize*, se etiqueta, desambigua, entre otros procesos.

Sí bien se logró implementar la herramienta y validarla, los modelos presentados inicialmente (modelos 00 y 01) no lograron cumplir el objetivo de encontrar los datasets relevantes en el Top-1, al cambiar de enfoque contando las palabras diferentes encontradas (modelos 02, 03 y 04) esto si resulto. Además tomar en cuenta que si bien el modelo con mayor precisión (modelo 02) no es el modelo con una mayor cantidad de procesos, esto nos dice que no necesariamente el último modelo será el mejor en este caso. Por otro lado existe el problema de que los requerimientos de información que poseen un valor en el campo *dataset_link* son una cantidad muy baja (8 requerimientos en total), lo que sería el 1% de los que existen en conjunto en todo el repositorio de datos abiertos. Estos resultados no son representativos para dar un veredicto si la herramienta desarrollada es satisfactoria o no.

7.2 Trabajo Futuro

En este proyecto se presentaron varias limitantes a lo largo de su desarrollo entre ellas se encuentra:

- El idioma
- La completitud de los datos del dataset

Algunos planes a futuro pueden ser:

- El idioma en el que se encuentran los datos, si bien es posible realizar un proceso de traducción esto llevaría un mayor tiempo de desarrollo, esto podría facilitar el acceso a usuarios de distintos idiomas acceder a sus datos.
- Dentro del proyecto se utilizó modelo booleano con algunos campos de los datasets presentados, esto porque la gran mayoría no poseía valores en ellos. Un aspecto importante a considerar a futuro sería, que los datasets idealmente se encontrara con un formato establecido y además con todos los valores de sus campos completos.
- Otro aspecto a considerar es agregar funcionalidades que sean capaces de detectar los campos de los metadatos que poseen un valor del dataset, éstos para poder usarlos en el *matching*, porque no todos los campos de los repositorios poseen valores, como se vio en este proyecto. Aumentando así la precisión de esta herramienta.
- Idealmente una vez obtenido el dataset más relevante se espera que a futuro la generación automática del dataset resultante. Ejemplo si los datos necesarios se encontraran en dos dataset diferentes, sería ideal que se creara un tercero (que contenga los datos de los dos datasets) de manera que se pueda responder esto de manera más sencilla que recuperando 2 datasets o más.

7.3 Anexos

7.3.1 Anexo A: Listado campos Requerimientos de Información

En el presente anexo se presenta un listado completo de todos los campos presentes en los requerimientos de información del portal de datos abiertos del Reino Unido.

Tabla 7.1: *Campos Requerimientos de Información*

Numero	Campo
1	Nid
2	URL
3	submitted_by
4	created
5	updated
6	data_themes
7	status
8	description
9	suggested_use
10	suggested_use_detail
11	benefits_overview
12	publisher
13	dataset_link
14	field_review_notes

7.3.2 Anexo B: Listado Campos Datasets

Tabla 7.2: *Listado Campos Datasets*

Numero	**Campo**
1	Name
2	Title
3	URL
4	Organization
5	Top level organisation
6	License
7	Published
8	NII
9	Location
10	Import source
11	Author
12	Geographic Coverage
13	Isopen
14	License Id
15	Maintainer
16	Mandate
17	Metadata Created
18	Metadata Modified
19	Notes
20	Odi Certificate
21	ODI Certificate URL
22	Tags
23	Temporal Coverage From

Continuación en la hoja siguiente

Tabla 7.2 – *Continuación de la hoja anterior*

Numero	**Campo**
24	Temporal Coverage To
25	Primary Theme
26	Secondary Themes
27	Update Frequency
28	Version

7.3.3 Anexo C: Listado Etiquetado Gramatical

En el presente anexo se presenta un listado completo de todas las posibles etiquetas gramaticales presentes en la función *pos_tag* de NLTK.

Tabla 7.3: *Listado Etiquetado Gramatical*

Number	**Tag**	**Description**
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential there
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun,plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb,comparative

Continuación en la hoja siguiente

Tabla 7.3 – *Continuación de la hoja anterior*

Number	**Tag**	**Description**
22.	RBS	Adverb,superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	to
26.	UH	Interjection
27.	VB	Verb,base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

Referencias

[1] E. L. Steven Bird Ewan Klein, *Natural language processing with Python*. O'Reilly, 2009.

[2] N. Hardeniya, *NLTK essentials: build cool NLP and machine learning applications using NLTK and other Python libraries*, P. Publ, Ed. 2015.

[3] J. Perkins, *Python 3 text processing with NLTK 3 cookbook: over 80 practical recipes on natural language processing techniques using Python's NLTK 3.0*, 2. ed, P. Publ, Ed. 2014.

[4] T. Berners-Lee, *Open data*, Jul. 2006. [Online]. Available: `https://www.w3.org/DesignIssues/LinkedData.html`.

[5] H. S., "Corpus linguistics," *Linguistics*, vol. 7, pp. 215–244, 2006.

[6] I. Nitin and D. F. J., *Handbook of natural language processing*, Second Edition, R. Herbrich and T. Graepel, Eds. 6000 Broken Sound Parkway NW, Suite 300: CRC Press, 2010, vol. 2.

[7] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques*, Third Edition. Elsevier, 2011.

[8] A. Zheng, "Evaluating machine learning models a beginner's guide to key concepts and pitfalls," 2015.

[9] T. Universität, D. University, of Lille, T. Universität, and Darmstadt, "Querying source code with natural language markus kimmig martin monperrus mira mezini," Tech. Rep., 2011.

[10] I. Habernal and M. Konopık, "SWSNL: Semantic web search using natural language," *Expert Systems with Applications*, vol. 40, no. 9, pp. 3649–3664, Jul. 2013. DOI: 10.1016/j.eswa.2012.12.070. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2012.12.070.

[11] M. A. Paredes-Valverde, M. Á. Rodrıguez-Garcıa, A. Ruiz-Martınez, R. Valencia-Garcıa, and G. Alor-Hernández, "ONLI: An ontology-based system for querying DBpedia using natural language paradigm," *Expert Systems with Applications*, vol. 42, no. 12, pp. 5163–5176, Jul. 2015. DOI: 10.1016/j.eswa.2015.02.034. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2015.02.034.

[12] N. Papadakis, P. Kefalas, and M. Stilianakakis, "A tool for access to relational databases in natural language," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7894–7900, Jun. 2011. DOI: 10.1016/j.eswa.2010.12.100. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2010.12.100.

[13] M. Llopis and A. Ferrández, "How to make a natural language interface to query databases accessible to everyone: An example," *Elsevier*, pp. 470–481, Oct. 2012.

[14] M. Yahya, K. Berberich, S. Elbassuoni, M. Ramanath, V. Tresp, and G. Weikum, "Natural language questions for the web of data," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, ser. EMNLP-CoNLL '12, Jeju Island, Korea: Association for Computational Linguistics, 2012, pp. 379–390. [Online]. Available: http://dl.acm.org/citation.cfm?id=2390948.2390995.

[15] S. Shekarpour and S. Auer, ""sina: Semantic interpretation of user queries for question answering on interlinked data" by saeedeh shekarpour with prateek jain as coordinator," *SIGWEB Newsl.*, no. Summer, 3:1–3:1, Jul. 2014, ISSN: 1931-1745. DOI: 10.1145/2641730.2641733. [Online]. Available: http://doi.acm.org/10.1145/2641730.2641733.

[16] D. Using and N. Language, "Freya: An interactive way of querying linked," in *ESWC 2011 Workshops*, R. Garcıa-Castro *et al.*, Eds., ser. LNCS, vol. 7117, Springer, 2012, pp. 125–138.

[17] N. D. Abdelghani Bouziane Djelloul Bouchiha and M. Malki, "Question answering systems: Survey and trends," *Procedia Computer Science*, vol. 73, pp. 366–375, 2015.

[18] V. S, S. Aithal, and P. Desai, "An approach towards automation of requirements analysis," in *Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS 2009, March 18 - 20, 2009, Hong Kong*, 2009.

[19] P. R. Pete Sawyer and R. Garside, "Revere: Support for requirements synthesis from documents," *Information Systems Frontiers*, 2002.

[20] C. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge University Press., 2009.