Universidad del País Vasco
Euskal Herriko Unibertsitatea

Tesis Doctoral

Traductor de consultas SPARQL, formuladas sobre fuentes de datos incompletamente alineadas, que aporta una estimación de la calidad de la traducción

---

Autora:
Ana I. Torre Bastida

Directores:
Prof. Dr. Jesús Bermúdez
Prof. Dr. Arantza Illarramendi

Memoria presentada cumpliendo los requisitos para optar al grado de

Doctor en Ingeniería Informática

en el

Departamento de Lenguajes y Sistemas Informáticos

Donostia, Febrero 2019

Cada día sabemos más y entendemos menos.

Albert Einstein

UNIVERSIDAD DEL PAÍS VASCO UPV/EHU

Resumen

Facultad de Informática (UPV/EHU)
Departamento de Lenguajes y Sistemas Informáticos

Doctor en Ingeniería Informática

Traductor de consultas SPARQL, formuladas sobre fuentes de datos incompletamente alineadas, que aporta una estimación de la calidad de la traducción

por Ana I. Torre Bastida

Hoy en día existe en la Web un número cada vez mayor de conjuntos de datos enlazados de distinta procedencia, referentes a diferentes dominios y que se encuentran accesibles al público en general para ser libremente explotados. Esta tesis doctoral centra su estudio en el ámbito del procesamiento de consultas sobre dicha nube de conjuntos de datos enlazados, abordando las dificultades en su acceso por aspectos relacionados con su heterogeneidad. La principal contribución reside en el planteamiento de una nueva propuesta que permite traducir la consulta realizada sobre un conjunto de datos enlazado a otro sin que estos se encuentren completamente alineados y sin que el usuario tenga que conocer las características técnicas inherentes a cada fuente de datos. Esta propuesta se materializa en un traductor que transforma una consulta SPARQL, adecuadamente expresada en términos de los vocabularios utilizados en un conjunto de datos de origen, en otra consulta SPARQL adecuadamente expresada para un conjunto de datos objetivo que involucra diferentes vocabularios. La traducción se basa en alineaciones existentes entre términos en diferentes conjuntos de datos. Cuando el traductor no puede producir una consulta semánticamente equivalente debido a la escasez de alineaciones de términos, el sistema produce una aproximación semántica de la consulta para evitar devolver una respuesta vacía al usuario. La traducción a través de los distintos conjuntos de datos se logra gracias a la aplicación de un variado grupo de reglas de transformación. En esta tesis se han definido cinco tipos de reglas, dependiendo de la motivación de la transformación, que son: equivalencia, jerarquía, basadas en las respuestas de la consulta, basadas en el perfil de los recursos que aparecen en la consulta y basadas en las características asociadas a los recursos que aparecen en la consulta.

Además, al no garantizar el traductor la preservación semántica debido a la heterogeneidad de los vocabularios se vuelve crucial el obtener una estimación de la calidad de la traducción producida. Por ello otra de las contribuciones relevantes de la tesis consiste en la definición del modo en que informar al usuario sobre la calidad de la consulta traducida, a través de dos indicadores: un factor de similaridad que se basa en el proceso de traducción en sí, y un indicador de calidad de los resultados, estimado gracias a un modelo predictivo. Este modelo se basa en técnicas de aprendizaje automático, y se entrena mediante un histórico compuesto por conjuntos de consultas SPARQL y sus traducciones previstas y validadas como correctas por un conjunto de expertos (gold standards).

Finalmente, esta tesis aporta una demostración de la viabilidad de la propuesta presentada a lo largo de todo el trabajo. Para lograrlo se ha establecido un marco de evaluación sobre el que se ha validado un prototipo del sistema. Dicho marco de evaluación se compone de un conjunto de preguntas recolectadas a partir de logs reales de puntos finales SPARQL y otros conocidos juegos de pruebas de referencia, además de métricas que actúan como indicadores para el análisis de la efectividad y rendimiento del prototipo. Los resultados obtenidos indican que el traductor efectivamente logra traducciones adecuadas e interesantes para los usuarios.

Agradecimientos

En primer lugar, me gustaría agradecer a mis dos directores de tesis, Jesús Bermúdez y Arantza Illarramendi, la oportunidad de doctorarme bajo su supervisión, y por brindarme durante todos estos años su experiencia y paciencia. Con ellos he aprendido a disfrutar la investigación y me he beneficiado de sus conocimientos, y su esfuerzo en enseñar a pensar. Aprecio especialmente a ambos por ser estupendas personas además de grandes educadores e investigadores. Su apoyo y estimulo han sido cruciales para que no desistiera en los momentos más difíciles.

En segundo lugar, a todos los compañeros de trabajo que se han cruzado en mi trayectoria durante estos 7 años, quiero agradecerles el ambiente de compañerismo y las grandes ideas que me han ido aportando. En todo este tiempo en la Universidad del Pais Vasco, he conocido a excelentes profesores y disfrutado de grandes compañeros. Especialmente quiero acordarme de mi grupo de investigación, BDI, y de mis compañeros del máster. A Tecnalia he de agradecerle el concederme la beca para empezar a desarrollar este doctorado, y el permitirme trabajar, conocer y compartir mi tiempo con personas increíbles, primero en el área de Infotech y ahora en el área de Optima. En esta organización he podido formar parte de iniciativas como el JRL, promovida y mantenida por formidables investigadores, que siempre están ahí para ayudarte. Y por último un especial reconocimiento a mis compañeros de fatigas, aquellas personas que han traspasado la barrera de compañeros para compartirse en amigos, todos esos investigadores que saben la carrera de fondo que supone el doctorado y que siempre me animaban a seguir. No quiero olvidarme de nadie así que no citaré nombres concretos, han sido muchas personas extraordinarias las que me he encontrado en este largo camino y todas me han ayudado en algún momento.

También tengo que acordarme de los amigos de toda la vida, la cuadrilla. Gracias por, aun no entendiendo muchas veces los temas de los que hablaba, seguirme animando y mostrando interés por como lo llevaba. Siempre estáis hay y para describiros solo puedo citar la siguiente frase:

> Todo mi patrimonio son mis amigos.
>
> Emily Dickinson

A mi familia, mi apoyo constante, los que me quieren sin doctorado y con doctorado, pero que entienden mi necesidad de nuevas metas y me animan a seguirlas, aunque me vean sufrir. Ellos son los que me enseñaron los pilares de un buen trabajo. A mis padres les agradezco estar siempre a mi lado y ser mis mejores críticos. Y a mis hermanas, por ayudarme a aprender lo que es un buen equipo y a luchar por lo que quieres.

Y por último, a mi compañero, que se ha convertido en mi marido durante este doctorado, solo puedo decirle que aún le quiero más por aguantarme y apoyarme en todo este tiempo. Un abrazo tuyo me calma, y este doctorado ha necesitado de muchos abrazos.

Índice

Lista de figuras

Lista de tablas

A mi familia y amigos, por vuestro apoyo

Capítulo 1

Introducción

Hoy en día nos encontramos en un mundo totalmente digitalizado, donde el número de sensores, dispositivos electrónicos y usuarios digitales aumenta cada segundo. Cada proceso digital o intercambios de información que estos producen genera, a su vez, una cantidad ingente de datos que se va sumando a la nube de datos ya existente. Al mismo tiempo, durante los últimos años, el gran crecimiento y auge de las aplicaciones disponibles en internet (geo-referencia, redes sociales, web 2.0) han contribuido a aumentar aún más el tamaño de esta nube de datos. Y, por último, la información almacenada en formato electrónico está comenzando a sobrepasar la almacenada en formato papel, que lleva ya más de una década decreciendo [HB11]. "Big Data", "Inundación de datos" o "sobrecarga de información" son solo algunas de las expresiones que se utilizan para describir la increíble e inmanejable cantidad de información disponible en Internet. Y la Web suele ser el mecanismo mediante el cual se accede a este compendio de información digital que está alcanzando dimensiones sin precedentes en términos de volumen, ámbito y accesibilidad.

La World Wide Web, coloquialmente Web, se puede describir como "un espacio de intercambio de información, en el que los elementos de interés se encuentran interconectados entre sí" [BLBC+04]. En un principio se trataba principalmente de un sistema de documentos de hipertexto interconectados, que podían ser de diversos formatos: texto, imágenes, videos y otros elementos multimedia. Por lo tanto, los documentos eran la columna vertebral de dicha Web clásica y para su especificación se utilizaba y utiliza el lenguaje HTML [RLHJ+99], que tiene como objetivo principal la representación y visualización de documentos. Por ejemplo, para recuperar la información relativa a la actriz Glenn Close, específicamente su ciudad de nacimiento, podemos acceder al documento HTML contenido en la URL `https://es.wikipedia.org/wiki/Glenn_Close` y analizarlo para descubrir los datos relevantes que nos interesen. Puede verse en la figura 1.1, donde hemos recuadrado en rojo la ciudad de nacimiento para destacar que es solo una parte más del texto, lo que complica su recuperación como información.

Según aumenta el volumen de datos en bruto y la necesidad de los usuarios de acceder a ellos, se acrecienta el interés de complementar la Web de documentos con una Web de datos. La gente demanda una nueva evolución de la Web, en la que sea más sencillo el acceso a su contenido, ya

FIGURA 1.1: Ejemplo de documento HTML en la Web con información sobre la actriz Glenn Close

que ahora hay una nueva comprensión del valor de los datos que contiene: "No son los documentos, son las datos y conceptos sobre los que tratan los que son importantes", como remarca Tim Berners-Lee, en el artículo de *Giant Global Graph* [BL13]. Y a medida que esta concepción de la Web como repositorio de datos se hace más popular, la utilización de formatos para estructurar la ingente vorágine de datos y el desarrollo de mecanismos para su consulta se vuelve crucial.

La figura 1.2 presenta un esquema conceptual de la Web clásica basada en documentos y cómo se puede extender hacia la Web actual basada en datos. En la primera parte de la imagen son los distintos documentos los que se encuentran vinculados unos con otros mediante enlaces. Sin embargo, en la Web de Datos (parte derecha de la imagen), el aspecto innovador y relevante lo encontramos en el posible enlazado de los recursos mediante propiedades etiquetadas, con las que se posibilita una descripción más detallada. Podemos resumir esta nueva concepción en la siguiente frase: "La Web de datos es la colección global de datos producidos por exposición y publicación sistemática y descentralizada de datos (en bruto) usando protocolos web." [Biz09].

Teniendo en cuenta que los datos se han convertido en la nueva unidad de información básica, y que es crucial estructurar esos datos para conseguir un acceso más eficiente, la Web convencional presenta dos carencias importantes:

1. Primero, sus documentos no se encuentran expresados en un formato legible por las máquinas. Si los datos y la información escalan a niveles que superen la capacidad humana, la única posibilidad de acceder, organizar y gestionar dichos datos es a través de máquinas.

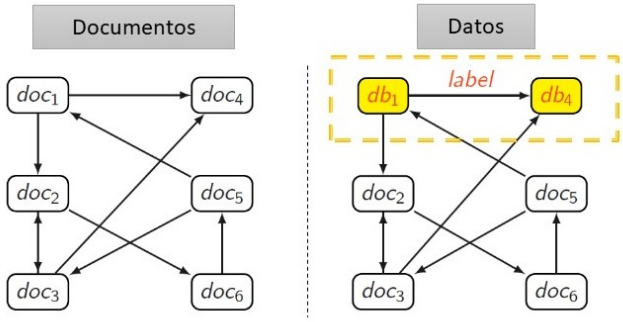


FIGURA 1.2: Comparativa entre Web de Documentos y Web de Datos.

Los datos legibles por máquina (datos automatizados) son datos almacenados en un formato que las máquinas pueden llegar a comprender, lo que permite a los agentes de software automatizados acceder a ellos y procesarlos sin intervención humana. Para los navegadores, los documentos web se componen solo de datos legibles por humanos. Aunque la mayoría de los sitios web tienen algún grado de estructura, el lenguaje en el que se crean, HTML, está orientado para estructurar documentos textuales en lugar de datos. Como los datos se encuentran dispersos en el texto que los rodea, es difícil para aplicaciones de software conseguir extraer fragmentos de datos estructurados de páginas HTML.

2. Segundo, sus documentos no se rigen por semántica alguna en la definición de su estructura y contenido. ¿Cómo podemos saber de forma inequívoca a qué se refiere o qué es lo que describe el texto, imagen o video contenido en un documento? ¿Cómo podemos comprender el significado de la información contenida en los documentos, si no utilizamos una semántica durante su definición? Nos referimos a dotar a los elementos que forman los documentos con una semántica mucho más sofisticada que la que pueden aportar los elementos de un lenguaje de marcado como es HTML.

Tomando como referencia el documento de la figura 1.1 estas dos carencias y los problemas asociados que conllevan, se aprecian fácilmente al utilizar un navegador para buscar documentos relativos a actores filtrándolos por su ciudad de nacimiento. Concretamente, en este caso buscaríamos actores que hubieran nacido en la ciudad de "Greenwich, Connecticut", y entonces el buscador nos devolvería, al menos, el documento que aparece en la imagen relativo a la actriz Glenn Close. Para llevar a cabo esta tarea, sería necesario que el buscador analizara cada documento de la Web para saber si su información es sobre un actor y si además en su contenido se encuentra el dato de la ciudad de nacimiento y coincide con el literal de

"Greenwich, Connecticut". Se puede intuir que la tarea es compleja, debido al gran número de documentos que el buscador debería procesar y a la cantidad de información que debería analizar por cada uno. Por lo tanto, podemos concluir que una búsqueda que a priori parece sencilla se vuelva compleja en la Web de documentos tradicional.

Para resolver estos problemas, los documentos Web convencionales pueden ser extendidos o anotados con datos adicionales para dotarlos de significado y estructura. Es la aproximación que defiende la Web Semántica, que se puede considerar la Web de datos procesable y legible por máquinas, y surge con el objetivo de organizar la información de la Web de Documentos. La Web Semántica debe considerarse una extensión mejorada de la Web clásica, citando a Tim Berners-Lee [BLHL01]:

> The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.
>
> —Tim Berners-Lee et al. [BLHL01], 2001

Para superar las dos carencias anteriormente presentadas, la Web Semántica se fija los siguientes dos objetivos:

1. Desarrollar lenguajes que permitan describir los metadatos necesarios de forma flexible, extensible y procesable por las máquinas. Se perfilan dos familias de lenguajes principales: a) Marco de descripción de recursos, RDF [CWL14] y b) lenguajes de definición de ontologías, principalmente OWL.

2. Desarrollar una infraestructura adecuada para implementar la nueva Web. Entre los bloques de tecnologías necesarios destacan: protocolos, el más común HTTP, lenguajes de consulta, como SPARQL [SWG13a] y especificaciones/paradigmas para acceder, consultar, publicar e intercambiar datos, como Linked Open Data [BHBL09].

En la tabla 1.1 se resumen las diferencias principales en tecnologías e infraestructuras entre la Web de documentos y la Web de datos. Destacan dos tecnologías que soportan el peso de las mejoras que supone la Web de Datos: RDF, para la especificación y SPARQL, para el acceso de los datos.

Tabla 1.1: Tabla de la evolución de la Web

	Web de Documentos Clásica	*Web de Datos actual*
Herramientas de acceso	Mecanismo de búsqueda	SPARQL, EndPoints
Lenguaje	HTML	RDF
Protocolo de acceso	HTTP	HTTP "++"
Datos primitivos	URI	IRI

1.1 RDF: Modelo de datos, sintaxis y esquema

A partir de aquí describimos de forma más detallada el pilar de la Web Semántica, y la tecnología capaz de aportar estructura a los datos: el modelo de datos RDF, que se introduce en W3C RDF Primer [MMM+04] y

se describe en detalle en [HPS14]. De ambos documentos es interesante subrayar la definición que se le da como: "Un lenguaje para la descripción de recursos en la World Wide Web", lo que nos aporta una noción de la relevancia que puede llegar a alcanzar este modelo en la Web de Datos para facilitar el procesamiento automático de estos. En él, la información se representa como grafos dirigidos etiquetados, donde los recursos son los nodos y las sentencias se forman estableciendo arcos (conexiones) entre dichos nodos. Está pensado para la representación e integración de información procedente de múltiples fuentes, por lo que es un compendio de diferentes esquemas. La potencia de RDF reside en la combinación de dos ideas:

- Un modelo flexible capaz de representar de manera uniforme tanto los datos simples como los metadatos asociados.
- Una estructura de grafo, que permite la representación de las interconexiones y relaciones entre los datos, de una forma natural y escalable.

El modelo RDF se basa en la identificación de los recursos con identificadores únicos, denominados IRIS (International Resource Identifiers) y en describir los recursos con la ayuda de propiedades simples y valores para dichas propiedades. IRIs son una generalización de URIs (Uniform Resource Identifiers), y son compatibles con estás y URLs. Los recursos en RDF pueden ser anónimos, es decir no tener una IRI asociada, y ser denominados como nodos en blanco. La descripción de un recurso se representa mediante un conjunto de sentencias, denominadas triples, porque su estructura refleja los tres componentes básicos de una oración: sujeto, predicado y objeto. El sujeto de un triple es la IRI que identifica al recurso descrito. El objeto puede ser un valor en forma de literal, como números, fechas o cadenas de texto, u otra IRI relacionada con el sujeto. El predicado es el encargado de establecer la relación existente entre el sujeto y el objeto, la propiedad.

Los triples RDF se pueden serializar mediante diferentes formatos, entre los que destacan RDF/XML[1], N3[2], N-triples[3] o Turtle[4]. En el listado 1.1 se muestran una serie de sentencias relacionadas con el dominio cinematográfico. Por ejemplo, el tercer triple describe al recurso *Glenn_Close* anotado bajo el espacio de nombre *dbr*, con la propiedad *starring* del espacio de nombres *dbo*, que tiene como objeto al recurso *Fatal_Attraction*.

```
@prefix db-r: <http://dbpedia.org/resource/> .
@prefix db-o: <http://dbpedia.org/ontology/> .
db-r:Glenn_Close
   db-o:birthDate
   "1947-03-19"^^<http://www.w3.org/2001/XMLSchema#date>;
   db-o:birthPlace db-r:Greenwich_Connecticut;
   db-o:starring db-r:Fatal_Attraction .
```

LISTADO 1.1: Ejemplo de triples RDF en sintaxis Turtle

[1] https://www.w3.org/TR/rdf-syntax-grammar/
[2] https://www.w3.org/TeamSubmission/n3/
[3] https://www.w3.org/TR/n-triples/
[4] https://www.w3.org/TR/turtle/

RDF schema (comúnmente abreviado como RDFS) [Bri04] es una extensión semántica de RDF. RDFS proporciona un vocabulario de modelado de datos para RDF, que provee de mecanismos para describir grupos de recursos relacionados (clases) y especificar las relaciones entre los recursos (propiedades). Según Brickley [Bri04] "consiste en una colección de recursos RDF que pueden usarse para describir otros recursos RDF en vocabularios RDF específicos de la aplicación". El vocabulario central al que hace referencia el autor se describe bajo el espacio de nombres informalmente denominado rdfs, y que se corresponde con la IRI `http://www.w3.org/2000/01/rdf-schema#`.

Los recursos se pueden distribuir en grupos llamados clases, por ello una clase en RDFS (rdfs:Class) corresponde a un concepto genérico de un tipo o categoría. Los miembros de una clase se conocen como instancias de la clase. Las clases son, en sí mismas, recursos. La propiedad rdf:type se usa para indicar que un recurso es una instancia de una clase, y se puede dar el caso de que una clase sea una instancia de sí misma. La semántica del modelo de datos RDF esta definida en [HPS14].

Una propiedad en RDFS (rdfs:Property) se describe como una relación entre el recurso del sujeto y el recurso del objeto. RDFS permite definir los tipos de valores que son apropiados para alguna propiedad, o en que clases tiene sentido atribuir tales propiedades. La forma en la que una propiedad se relaciona con una clase, está regida principalmente por dos propiedades en RDFS: dominio (rdfs:domain) y rango(rdfs:range).

Otro aporte importante de esta especificación es la posible descripción de jerarquías de clases y propiedades. A continuación, incluimos la explicación aportada en la recomendación del W3C [Bri04]:

- La propiedad rdfs:subClassOf se puede usar para indicar que una clase es una subclase de otra. Si una clase C es una subclase de una clase C', entonces todas las instancias de C también serán instancias de C'. El término superclase se usa como el inverso de la subclase. Si una clase C' es una superclase de una clase C, entonces todas las instancias de C también son instancias de C'.

- La propiedad rdfs:subPropertyOf se puede usar para indicar que una propiedad es una subpropiedad de otra. Si una propiedad P es una subpropiedad de la propiedad P', todos los pares de recursos que están relacionados por P también están relacionados por P'. El término super-propiedad a menudo se usa como el inverso de la subpropiedad. Si una propiedad P' es una super-propiedad de una propiedad P, entonces todos los pares de recursos que están relacionados por P también están relacionados por P'.

Por lo tanto, RDF schema extiende a RDF para aportar un vocabulario con significado adicional. Esto vamos a intentar plasmarlo en la figura 1.3 siguiendo el mismo ejemplo utilizado hasta ahora. En él puede verse el grafo correspondiente a la estructura de datos de la figura 1.1 completado con el vocabulario RDFS, como rdfs:Class o rdfs:label. Concretamente se han definido las clases Actor y Film, para especificar el tipo de los recursos Glenn_Close y Fatal_Attraction, respectivamente. Y se ha utilizado la

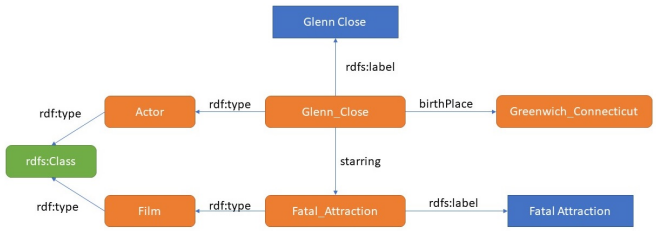


FIGURA 1.3: Estructura de datos en forma de grafo para el ejemplo de la figura 1.1.

propiedad rdfs:label cuyo valor es el literal con el que se nombran los dos recursos anteriores.

1.2 SPARQL

RDF permite modelar la Web de datos, pero para acceder a los datos y recuperar su información necesitamos un lenguaje de consulta capaz de trabajar sobre dicho modelo. El lenguaje estándar para consultar RDF es SPARQL [SWG13a], y se basa en la coincidencia de patrones de grafos RDF. Es decir, su implementación trabaja buscando coincidencias entre los patrones especificados en la consulta y los existentes en los grafos RDF de los conjuntos de datos enlazados. Los patrones de grafo más complejos se pueden formar a partir de la combinación en distintas formas de patrones más pequeños. Una definición formal de un patrón de grafo puede verse a continuación, y está detallada en el apéndice A, la sección A.3:

Definición 1 *Un patrón de grafo se define de forma recursiva como:*

- *Un patrón de grafo es un patrón de triple, donde cualquiera de los elementos s, p, o puede ser sustituido por una variable $?x$.*
- *Si P_1 and P_2 son patrones de grafo, entonces $(P_1\ AND\ P_2)$, $(P_1\ OPT\ P_2)$ y $(P_1\ UNION\ P_2)$ son patrones de grafo más complejos.*
- *Si P es un patrón de grafo, entonces $FILTER\ P$ es un patrón de grafo.*

Hay cuatro constructores básicos a especificar en una consulta SPARQL: Seleccionar (SELECT), Construir (CONSTRUCT), Preguntar (ASK) y Describir (DESCRIBE). Su principal diferencia reside en cómo se devuelve y construye el conjunto de resultados. El más utilizado y soportado por los diferentes mecanismos y motores de búsqueda basados en SPARQL es el constructor Select.

Además de un estándar para la consulta, SPARQL define un protocolo que especifica cómo ejecutar las consultas y recuperar los resultados. Por

ello, los administradores de los conjuntos de datos deben habilitar interfaces que permitan la consulta en formato SPARQL, son los conocidos puntos finales de acceso (SPARQL Endpoint).

Definición 2 *Un punto final SPARQL es un servicio de protocolo SPARQL definido conforme a la especificación SPROT. Un punto final SPARQL permite a los usuarios (humanos u otros) consultar una base de conocimiento a través del lenguaje SPARQL. Los resultados se devuelven típicamente en uno o más formatos procesables por máquina.*

—*SPARQL Query Language for RDF*[5], *2008*

En cuanto a la evolución y estado actual del lenguaje existen dos versiones:

- 1.0: Recomendación del W3C y soportada por la mayoría de puntos finales de acceso.
- 1.1: Es la recomendación más reciente (marzo 2013), solo se encuentra soportada por una minoría de puntos finales de acceso.

Hoy en día la mayoría de puntos de acceso habilitados implementan únicamente la versión 1.0, por lo tanto, es la versión que se utilizará en este trabajo. En la consulta inferior se recuperan las ciudades del condado en el que nació la actriz Glenn Close, sobre el conjunto de datos DBpedia y utilizando dos vocabularios: la ontología DBpedia y la ontología Geonames[6]:

```
SELECT ?ciudades
WHERE {
<http://dbpedia.org/resource/Glenn_Close>
<http://dbpedia.org/ontology/birthPlace> ?l .
?l <http://www.geonames.org/ontology#parentFeature>
?parentPlace .
?parentPlace
<http://www.geonames.org/ontology#nearbyFeatures>
?ciudades .
}
```

1.3 Linked Open Data - Datos abiertos y enlazados

En general, se puede considerar RDF como la infraestructura y base tecnológica necesaria para comenzar a desarrollar la idea de una Web de Datos, y SPARQL como el lenguaje para consultarla, pero siguen existiendo dos problemas importantes:

- Por una parte, buscar la solución al problema de la distribución de los datos en la Web, con la existencia de múltiples orígenes y la consecuente necesidad de relacionar unos con otros.

[5] https://www.w3.org/TR/rdf-sparql-query/
[6] http://www.geonames.org/ontology/documentation.html

- Y por otra, la necesidad de impulsar la compartición y accesibilidad de la información, ya que para conseguir que los datos sean fácilmente recuperables es básico promover la utilización de la Web como un espacio de información global.

El reciente surgimiento del enfoque de Datos Abiertos Vinculados - Linked Open Data (LOD) [BHBL09] para publicación de datos en abierto representa un gran paso adelante en la superación de estas dos barreras y en la consecución de la visión original de Berners-Lee, Hendler y Lassila de la Web Semántica como un espacio global abierto a la compartición de información [BLHL01]. Técnicamente, LOD define un paradigma para la publicación y el consumo del contenido que se incluye en la Web, es decir, un conjunto de mejores prácticas para publicar e interconectar datos estructurados en la Web y compartirlos de forma abierta. Citando al propio proyecto Linked Data:

> "Linked Data". Se trata de usar la Web para conectar datos relacionados que anteriormente no estaban vinculados. Para ello se recomiendan un conjunto de buenas prácticas para publicar, compartir y conectar datos, información y conocimiento en la Web Semántica usando URIs y el modelo RDF.
>
> —Linked Data Project[7], 2007

Los principios en los que se basa actualmente son:

- Usar IRI para referirse e identificar los recursos.
- Usar HTTP IRI para que los usuarios puedan acceder y detectar los identificadores.
- Proporcionar información útil, utilizando los estándares RDF y SPARQL.
- Incluir enlaces a otros IRI, para que más recursos puedan ser descubiertos.

Los datos vinculados basan su publicación en el uso de lenguaje RDF para su formato, IRIs referenciables para su identificación y el protocolo HTTP para su acceso. Con IRIs referenciables nos referimos a aquellas que diferencian correctamente entre la referencia del recurso y la descripción del recurso, de modo que el usuario tendrá acceso al primero (referencia) normalmente expresado en formato HTML y las máquinas accederán al segundo (descripción) para su procesamiento, generalmente en formato RDF. Siguiendo con el ejemplo utilizado hasta ahora sobre la actriz Glenn Close, en la figura 1.4 puede verse la combinación de documento HTML y fichero RDF (formato N-Triples) que forman la referencia y descripción del recurso, respectivamente.

Uno de los puntos fuertes y diferenciales del paradigma Linked Data es su afán en enlazar los recursos procedentes de los diferentes orígenes de datos de la Web. Para comprender por qué esta tarea es primordial,

[7] http://linkeddata.org/

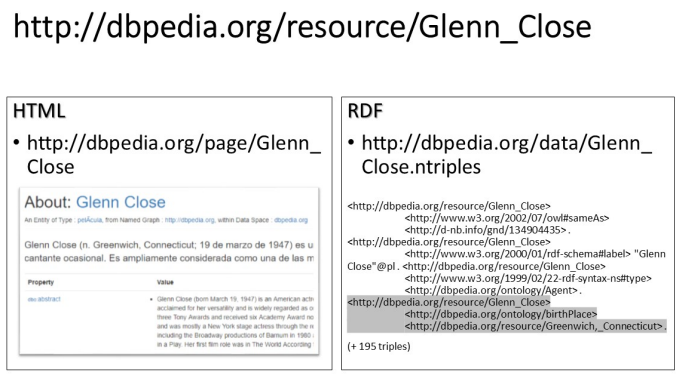


Figura 1.4: Combinación de RDF y HTML para la IRI referenciables del ejemplo de la figura 1.1.

debemos analizar primero el ciclo de vida que sigue todo dato enlazado [ABD+12], entendiéndolo como el proceso mediante el cual los datos captados desde los diferentes orígenes terminan formando parte de conjuntos de datos enlazados y públicos. Se puede resumir en tres fases:

1. Creación del conjunto de datos. En esta fase se produce la extracción de los datos de sus orígenes, la generación de las IRI HTTP para cada recurso y la selección del vocabulario.

2. Vinculación con el resto de conjuntos de datos. Supone descubrir y especificar las relaciones entre los diferentes recursos existentes en los conjuntos de datos.

3. Publicación del conjunto de datos. En esta fase es necesario permitir el acceso a los datos y asociar ciertos metadatos al conjunto de datos para facilitar su uso.

Al hablar de origen de datos nos referimos a la fuente de la que provienen los datos primitivos. En el caso de conjunto de datos consideramos una representación estructurada de datos (formato RDF) procedentes de un origen; lo que termina siendo una colección de descripciones de los recursos contenidos en los datos de dicho origen. Además, al tener la característica de enlazados, sus recursos pueden encontrarse vinculados con recursos de otros conjuntos de datos por medio de enlaces. A la hora de crear un conjunto de datos es necesario elegir un vocabulario con el que se anotan y describen los recursos. Una buena práctica es reutilizar vocabularios ya existentes utilizados para otros conjuntos de datos, de esta forma se evita la ambigüedad y se fomenta la uniformidad en las descripciones. Pueden usarse varios vocabularios para anotar los recursos de un único conjunto de datos. A cada vocabulario se le asigna un espacio de nombre que tiene un IRI de referencia, que se utiliza como base para la generación del resto de IRIs de los recursos incluidos en ese vocabulario.

En el estándar de ontologías publicado por el W3C [Sta18], se establece que, en la Web semántica, "los vocabularios definen los conceptos y las relaciones (también denominados "términos") que se utilizan para describir y representar un área de interés. Los vocabularios se usan para clasificar los términos que se pueden usar en una aplicación en particular, caracterizar las posibles relaciones y definir las posibles restricciones para usar esos términos." En este mismo documento comentan cómo no existe una división evidente entre lo que se conoce como "vocabularios" y "ontologías" , y presentan que la tendencia es usar la palabra "ontología" para una colección de términos más compleja y posiblemente más formal, mientras que se usa "vocabulario" cuando no es necesario un formalismo tan estricto o se utiliza de una forma relajada.

En el estándar del W3C plantean que la función de los vocabularios en la Web Semántica se centra en dos tareas cruciales, la inferencia de conocimiento y la integración de los datos. Estas dos funciones son la razón por la cual los vocabularios son un pilar en el desarrollo de esta tesis, ya que pueden solucionar la existencia de ambigüedades en los términos utilizados en los diferentes conjuntos de datos, y permiten el descubrimiento de nuevas relaciones a partir de conocimiento inferido de ellos. Las definiciones para los conceptos clave introducidos son las siguientes:

Definición 3 *Un conjunto de datos D es un conjunto de triples RDF.*

Definición 4 *Un vocabulario V es un conjunto de triples que describen clases C y propiedades P. Cualquier conjunto de datos D puede contener un triple t en cuya descripción $t = (s, p, o)$ el objeto o y el predicado p pertenecen a las clases y propiedades descritas en V, es decir $o \in C$ y $p \in P$.*

En la tabla 1.2 puede verse un ejemplo de conjunto de datos, DBpedia[8], uno de los vocabularios de datos utilizados para la anotación de sus recursos, el espacio de nombres que se ha asociado a dicho vocabulario, y un ejemplo de uno de los términos del vocabulario. En los conjuntos de datos, se pueden distinguir dos tipos de triples dependiendo del tipo de su objeto:

1. Triples Literales, son aquellos cuyo objeto es un literal RDF. Dichos literales sirven para describir las propiedades de los recursos. El primer triple incluido en el listado 1.1 es de este tipo y representa la fecha de nacimiento de la actriz Glenn Close. Los literales pueden ser simples o tipados.

2. Links RDF, permiten describir la relación entre dos recursos. El segundo triple incluido en el listado 1.1 es de este tipo y describe la relación entre la actriz Glenn Close y la ciudad de Greenwich como ciudad de nacimiento.

[8] https://wiki.dbpedia.org/

Tabla 1.2: Ejemplo Conjunto de datos

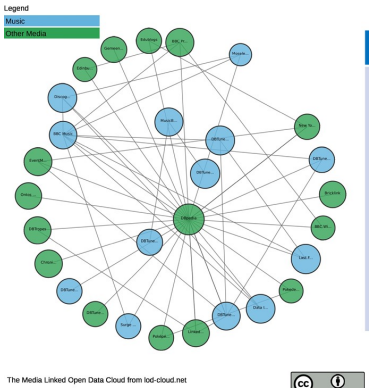
Conjunto de datos	DBpedia
Ejemplo vocabulario de datos	DBpedia Ontology (dbpedia-owl) The DBpedia ontology provides the classes and properties used in the DBpedia data set
Espacio de nombres	http://dbpedia.org/ontology/
Ejemplo de término	Clase Actor http://dbpedia.org/ontology/Actor

1.3.1 Problemas de enlazado y requisitos de la consulta en la LOD

El objetivo final del paradigma Linked Open Data es ser aplicado de forma global en la Web y lograr una arquitectura general capaz de compartir datos estructurados, hasta alcanzar un repositorio global único. Esta idea empieza a materializarse en la actual nube de datos abiertos y vinculados (Linked Open Data Cloud- LODC) y tiene como objetivo final alcanzar la implementación de la Web de Datos. Y es que el valor de publicar los datos en la Web reside principalmente en poder consultarlos en combinación con otros datos de los que en un principio ni siquiera se tenía por qué tener conocimiento. Por lo tanto, es posible imaginar todo el conjunto de datos vinculados existentes en la Web como un grafo global gigante, que concuerda con la visión, ya citada anteriormente, ofrecida por Tim Berners-Lee en [BL13]. Los usuarios pueden conectar datos científicos, datos comunitarios, sociales, empresariales o gubernamentales, además de datos procedentes de muy diversas organizaciones, agencias o incluso de diversos países. Esto se consigue gracias a que los datos se publican de forma descentralizada, es como una gran telaraña que se va tejiendo a medida que nuevos agentes publican sus datos. La predicción de Tim Berners-Lee se ve secundada por Jain et al. en el artículo [JHY+10], en el cual establecen cómo la LOD ofrece muchas posibilidades de ser el comienzo de la Web Semántica, pero también detectan problemas que pueden limitar la implementación de esta Web. Y es que esta forma de publicación, que en un principio aporta el gran beneficio de no necesitar de un punto central o mediador que podría ser engorroso, tiene también sus desventajas. Entre las limitaciones que identifican mencionamos las siguientes:

1. Las interconexiones actuales entre los conjuntos de datos en el LOD Cloud son demasiado superficiales y escasas, por lo que se corre el riesgo de no ofrecer más información que la actual Web de Documentos. Se está generando una red pobre de enlaces y conjuntos de datos parcialmente vinculados, lo que en esta tesis se ha denominado como conjuntos de datos "incompletamente" alineados.

2. Falta de descripciones conceptuales sobre los orígenes de datos y el tipo de datos que producen, lo cual no favorece su descubrimiento o utilización. Los autores del artículo [JHY+10] explican cómo a pesar de que se han hecho esfuerzos por idear soluciones para describir los conjuntos de datos, como la presentada en el artículo de Alexander et al. [AH09], estos trabajos se centran más en aspectos estadísticos de



Musical	Otros dominios audiovisuales
DBTune	Dbpedia
MusicBrainz	New Tork Times
Moseley Folk Festival Data	BrickLink
Discogs in RDF	BBC
BBC Music	Pokedex
Surge Radio	Linked Movie DataBase
Data Incubator: MusicBrainz	Poképédia
Last.FM RDFization	Chronicling America
	DBTropes
	Ontos News Portal
	EventMedia
	Edinburgh University
	Theatre Company
	Productions
	Gemeenschappelijke
	Edublogs
	BBC Programmes

FIGURA 1.5: Sub-nube de conjuntos de datos del dominio audiovisual.

los conjuntos de datos y no se ajustan a los requisitos para capturar información conceptual.

3. Dificultades durante la consulta, relacionadas con: la heterogeneidad en la forma de modelar los datos dependiendo de su origen, la declaración ambigua de entidades entre los diferentes orígenes, y por último la puntuación o ranking de la información dependiendo del origen del que proceda.

Nosotros resumimos los problemas anteriores en que la estructura altamente distribuida y la naturaleza evolutiva del entorno junto con el alto número de fuentes de datos hace que para los usuarios se convierta en una tarea muy compleja conocer la ubicación, la estructura, el vocabulario y la semántica de cada conjunto de datos en particular. Y el abordar esas dificultades detectadas durante la consulta es la principal motivación de esta tesis.

Como ejemplo, imaginemos que el usuario desea recopilar todos los títulos de las canciones de las diferentes bandas sonoras de películas en las que ha aparecido la actriz Glenn Close. Los posibles conjuntos de datos que pueden ser relevantes para la consulta pueden verse en la figura 1.5, que representa la sub nube de conjuntos de datos relacionada con el dominio audiovisual (principalmente música y cine); ejemplos principales son la DBpedia, MusicBrainz, DBTune o Linked Movie Database, entre muchos otros. A partir de este grafo podemos imaginar que el conocer los términos de todos los vocabularios para los diferentes conjuntos de datos mostrados, de forma que nos resulte sencillo formular la pregunta planteada, es una tarea muy costosa y que requiere de un conocimiento experto y años de experiencia, además de tener que estar continuamente pendiente de las posibles modificaciones a los vocabularios y nuevos datos.

Como hemos ido viendo a lo largo de toda esta introducción, los enlaces son fundamentales para la Web de datos, ya que son el "instrumento" que permite conectar los conjuntos de datos aislados formando un repositorio global. Debido a la heterogeneidad de los conjuntos de datos y a

los diferentes vocabularios utilizados para modelar sus datos, los procesos dedicados a identificar las relaciones entre los diferentes recursos existentes en los conjuntos de datos y establecer enlaces basándose en ellas son tareas complejas y críticas. Podemos distinguir dos procesos principales en la tarea de relacionar recursos, tal como se presenta en Euzenat et al. [ES13a].

- Alineamiento entre vocabularios (Ontology matching and alignment). La red de datos abarca una amplia gama de dominios y sus conjuntos de datos se definen mediante diferentes vocabularios/ontologías asociadas a estos dominios, que a veces pueden superponerse o complementarse. Por lo tanto, cuando trabajamos con todos los conjuntos de datos, necesitamos procesos (basados en técnicas de coincidencia de ontología) que nos permitan identificar cuándo términos pertenecientes a diferentes vocabularios denotan el mismo recurso en la vida real.

- Enlazado de datos (Data Interlinking). El proceso de identificar la misma entidad en diferentes conjuntos de datos y publicar un enlace entre ellas.

A continuación, mostramos un ejemplo de enlace, entendido este como triple RDF en el que la propiedad (en este caso de equivalencia, owl:sameAs) refleja la relación entre el sujeto y el objeto. El ejemplo enlaza dos recursos que referencian a la misma entidad real, la actriz Glenn Close, y que se encuentran definidos en dos conjuntos de datos diferentes que aparecían en la sub-nube de la figura 1.5 como son la DBpedia y MusicBrainz.

```
<http://dbpedia.org/resource/Glenn_Close>
owl:sameAs
<http://musicbrainz.org/artist/
512f6db1-63ef-42d1-a234-55bacecfe0c0> .
```

En el trabajo previamente citado [JHY+10], los autores enuncian lo que ellos consideran la solución a los problemas que ya hemos explicado sobre la heterogeneidad de vocabularios en la LODC, basándose en dos propuestas: el desarrollo de una ontología de alto nivel, que ofrezca un modelo global, y un mecanismo para consultar de forma eficiente la nube de conjuntos de datos enlazados. La opción de utilizar una ontología de alto nivel como mediadora en el modelo de los datos, nos parece que obligaría a una mayor precisión y conocimiento ontológico en la tarea de anotación de recursos, lo que en cierta manera podría desmotivar a los publicadores ya que el esfuerzo necesario es mayor. Además, los agentes publicadores de los datos se verían obligados a estructurar sus datos siguiendo dicha ontología y perderían la capacidad de participar en la definición del modelo de forma activa. Sin embargo, estamos de acuerdo con su segunda propuesta ya que, en este escenario se hace imprescindible proporcionar a los usuarios herramientas y mecanismos que los ayuden a explotar la gran cantidad de datos disponibles de forma óptima, y esta es la base para la propuesta presentada en esta tesis. A continuación listamos los requisitos y características que planteamos como prioritarios para un sistema de consulta que implemente dicha propuesta:

1. Facilidad para expresar consultas. El sistema debe ofrecer a los usuarios algunas de las siguientes facilidades cuando consultan conjuntos de datos heterogéneos de la LOD: uso de palabras clave, expresión de la consulta usando lenguaje natural o patrones predefinidos, o expresión de la consulta utilizando el vocabulario con el que los usuarios están familiarizados.

2. Manejo de vocabularios heterogéneos a partir de conjuntos de datos no preseleccionados. El sistema debe ser capaz de proporcionar respuestas accediendo a conjuntos de datos que usan diferentes vocabularios.

3. Sin necesidad de preprocesamiento. El sistema no debe requerir de costosas tareas de preprocesamiento dedicadas a crear un repositorio centralizado, a construir una federación de conjuntos de datos, o un índice específico. Es decir no debe basarse en la necesidad de realizar tareas previas de integración.

4. Respuestas imprecisas. La admisión de respuestas no exactas en la sintaxis y semántica para favorecer la obtención de respuestas es aceptable.

5. Admisión de la expresividad SPARQL completa. El sistema no debe imponer restricciones durante la formulación de la consulta SPARQL.

1.4 Enfoques y escenarios de consulta en el entorno de la Linked Open Data

Remarcando que la potencia y valor de la Web de Datos reside en la posibilidad de consulta sobre datos de muy diversos orígenes, y el punto establecido anteriormente de la urgente necesidad de desarrollar nuevos mecanismos para consultarla de forma eficiente vamos a repasar y analizar los diferentes enfoques de consulta ya existentes. La intención es valorar los avances existentes en la literatura sobre procesamiento de consultas en los conjuntos de datos enlazados, lo que hemos denominado enfoques de consulta siguiendo la nomenclatura propuesta en el trabajo de Hartig et al. [HBF09].

A continuación, describiremos los diferentes enfoques de procesamiento de consulta, todos ellos tienen un objetivo similar: analizar la Web de Datos mediante consultas, pero difieren en el alcance, la implementación, la complejidad y el rendimiento de la tarea. Vamos a seguir la siguiente clasificación para describirlos acorde al trabajo de Hartig et al. [HBF09]: consulta individual de un conjunto de datos enlazados, enfoque centralizado denominado datawarehouse, enfoque federado y enfoque exploratorio. Utilizaremos el ejemplo, ya anteriormente presentado, para ilustrar sus principales limitaciones:

Consideremos una consulta en la que el usuario quiere conocer los distintos trabajos en los que ha participado la artista Glenn Close.

1. **Consulta uno a uno de los conjuntos de datos LOD.** Es el caso más simple, donde una aplicación o usuario puede consultar un conjunto de datos a través del punto final de acceso habilitado por el publicador del conjunto de datos. Este enfoque tiene la limitación de ignorar el potencial de la Web de Datos, el usuario final tiene que ser consciente de la distribución de los datos entre los diferentes orígenes, luego no puede considerarlo un repositorio global.

   En este caso la consulta anterior no puede responderse accediendo a un único conjunto de datos, debido a que la información para la respuesta está localizada en varios. La artista se considera actriz y cantante, por lo que abarca conjuntos de datos del dominio del cine y la música, respectivamente. La obra musical de la artista puede recuperarse accediendo a MusicBrainz (dominio musical). Pero la información sobre las películas en las que participó se encuentra en un conjunto de datos especializado en el dominio del cine como es LinkedMDB (dominio cinematográfico).

2. **Enfoque centralizado - Datawarehouse.** Este método implica que varios conjuntos de datos se recopilan de antemano, se preprocesan y se almacenan en repositorios únicos centralizados y las consultas se evalúan en los puntos de acceso habilitados en dichos repositorios. Los inconvenientes son dos principalmente: a) el coste de configurar, gestionar y mantener un repositorio centralizado y b) la desincronización, los datos pueden quedar obsoletos o la respuesta a la consulta puede no ser completa, en relación al estado actual del conjunto de datos original. En el ejemplo, para implementar este enfoque sería necesario integrar los conjuntos de datos: LinkedMDB y MusicBrainz. Esto conlleva dos desventajas asociadas, primero el coste de generación y segundo, en el caso de que se produjeran, por ejemplo, cambios en el nombre de alguna de las obras musicales en el conjunto de datos MusicBrainz, o adicciones de nuevas películas en el conjunto de datos LinkedMDB, se originaría un periodo de tiempo en el que los datos del repositorio integrado estarían inconsistentes o incompletos, respectivamente.

3. **Federación de consultas.** Los enfoques de procesamiento de consultas federadas distribuyen la ejecución de consultas sobre los puntos finales de acceso que los distintos conjuntos de datos habilitan al efecto. En la reciente literatura se han presentado una gran cantidad de sistemas de federación sobre conjuntos de datos enlazados (por ejemplo, ANAPSID [AVL+11], DARQ [QL08a], FedX [SHH+11a] y SPLENDID [GS11]). La ventaja de utilizar un sistema de este tipo es que no es necesario procesos de sincronización o consolidación sobre los datos, ya que las consultas siempre se responden sobre los datos originales y actualizados. Pero el utilizar un enfoque federado no evita al usuario tener que conocer el "vocabulario federado" adecuado para acceder a los diferentes conjuntos de datos y que se debe tener en cuenta en el momento de definición de la consulta. Si se utilizara este enfoque para la consulta del ejemplo, la solución

podría ser el desarrollo de una ontología federada para el sistema sobre los conjuntos de datos MusicBrainz y LinkedMDB, pero en este caso el usuario tendría que familiarizarse con esta nueva ontología. Además esta solución supone un esfuerzo adicional para los administradores del sistema federado que deberían mantener actualizada esta ontología acorde a los cambios que se produzcan en los vocabularios de los conjuntos de datos que lo conforman. Es decir, en el caso de usar una ontología federada para este tipo de sistemas no es necesario sincronizar los datos pero si reflejar los cambios y adiciones en la estructura y modelo de estos en dicha ontología.

4. **Procesamiento de consultas explorativo.** En este enfoque se aprovecha el principio de IRI referenciable promovido por los datos enlazados. La ejecución de consultas comienza en un conjunto de datos fuente y utiliza las IRI HTTP referenciables devueltas para recuperar más datos recorriendo los enlaces de estas, y de esa forma agregar datos adicionales para responder partes de la consulta e ir incluyendo más IRIs que pueden ser sucesivamente "exploradas" para aumentar el conjunto de resultados, hasta que la consulta inicial sea suficientemente respondida.

   Este enfoque permite aprovechar todo el potencial de la Web de datos pero, dependiendo de su implementación y alcance, puede ser muy costoso computacionalmente; el número de enlaces a recorrer puede ser ingente, y la precisión y legibilidad de las respuestas devueltas pueden verse mermadas por la ingente cantidad de datos que el usuario tiene que considerar. Además, depende completamente de las interconexiones entre los recursos que, como comentábamos, actualmente son superficiales y escasas. Volvamos de nuevo al ejemplo, si no hubiera enlaces entre la IRI que representa a la artista Glen Close en LinkedMDB y el que la representa en MusicBrainz, este enfoque sería imposible de desarrollar.

Todas las carencias presentadas en los enfoques anteriores se ven acrecientadas al consultar datos del mismo dominio, pero distribuidos en diversos orígenes, es decir conjuntos de datos que presentan una superposición. Esto es debido a la necesidad de que los mecanismos de consulta sean capaces de lidiar con el uso de vocabularios diferentes dependiendo del conjunto de datos al que se acceda para definir datos que representan lo mismo en el mundo real. Volvemos al ejemplo anterior, en él hemos utilizado los conjuntos de datos LinkedMDB (dominio cinematográfico) y MusicBrainz (dominio musical). Existen también conjuntos de datos de dominios múltiples (que ya hemos visto en la figura 1.5) como DBpedia y otros específicos del dominio audiovisual como BBC[9] que también contienen información sobre la artista Glenn Close. El problema es que esta entidad se encuentra anotada de diferente manera utilizando vocabularios que se superponen. Por ejemplo, para definir la clase Actor podemos ver en la tabla 1.3 las IRIs asociadas en los diferentes vocabularios utilizados por cada conjunto de datos.

[9] https://www.bbc.co.uk/things/

Tabla 1.3: Ejemplo de IRIs procedentes de diferentes vocabularios referidas a la clase Actor

	Clase Actor
DBpedia	<http://dbpedia.org/ontology/Actor>
LinkedMDB	<http://data.linkedmdb.org/resource/movie/actor>
BBC	<https://www.bbc.co.uk/things/0ca7e0fd-8b11-40af-a2f0-3f430a6d6146>

A partir de la definición de los enfoques generales de consulta, se pretende establecer e ilustrar una serie de posibles escenarios básicos. En ellos, se plasman problemas sin resolver a día de hoy y se muestra la necesidad de aportar un nuevo enfoque para la consulta, son los siguientes ejemplos:

- Un usuario común que plantea una consulta basada en palabras clave a un sistema de preguntas y respuestas (Question Answering - QA). El sistema deberá encargarse de generar la correspondiente consulta SPARQL para poder ejecutarla sobre el punto de acceso de un conjunto de datos elegido en un inicio por el usuario. Además, el sistema deberá ser capaz de validar si las respuestas son suficientes para el usuario y en caso negativo demandar respuestas adicionales en otros conjuntos de datos, que pueden estar definidos con vocabularios diferentes.

- Un científico acostumbrado a plantear consultas SPARQL sobre un conjunto de datos con el que se encuentra familiarizado, y que en ocasiones le ayudaría poder explorar, para una temática concreta en la que se le plantea una consulta, el resto de los conjuntos de datos disponibles en la Web de datos. El usuario comienza formulando una consulta exploratoria sobre el conjunto de datos de origen con el que se encuentra familiarizado, y le requiere al sistema que busque respuestas adicionales accediendo a otros conjuntos de datos. Un punto importante a tener en cuenta es que el científico no tiene conocimiento sobre la estructura/vocabulario de los diferentes conjuntos de datos que se vayan consultando a lo largo de la búsqueda.

- Un programador de aplicaciones, conocedor de los lenguajes de consulta como SPARQL y familiarizado con el vocabulario del conjunto de datos DBpedia. Su intención es utilizar la DBpedia para ampliar los resultados de ciertas consultas que le han sido dadas. Estas consultas han sido definidas para distintos conjuntos de datos por otros programadores expertos en los vocabularios adecuados. Por lo que, para obtener respuestas, es necesaria una traducción de la consulta de origen para poder expresarla adecuadamente en los términos en inglés de la DBpedia.

1.5 Motivación y descripción de la solución propuesta

En el contexto considerado no es sencillo para los usuarios tener el conocimiento sobre las características técnicas necesarias para realizar consultas

efectivas a los distintos conjuntos de datos, y menos aún llegar a ser expertos en el total de conjuntos de datos. El entorno es demasiado amplio y cambiante. Tampoco es fácil diseñar una herramienta que aúne y modele los diferentes conjuntos de datos enlazados, proporcionando una interfaz de acceso y consulta única. Principalmente porque la tarea de integración sería ingente y en un entorno en evolución como el que estamos requeriría de un mantenimiento continuo. En resumen, los enfoques federado y centralizado quedan descartados por las razones anteriores. Por tanto, el enfoque que parece más útil es la exploración incremental de los conjuntos de datos. Pero se aprecian dos inconvenientes: primero, el número de IRIs enlazadas entre conjuntos de datos es bajo, por lo que en ocasiones es necesario seguir cadenas de búsqueda muy profundas y esto puede hacer que la consulta sea computacionalmente muy costosa, ofrezca rendimientos muy pobres en tiempos y en ocasiones incluso no pueda proveer de resultados al usuario. En el ejemplo, imaginemos que no existe un link de enlace entre la entidad de la actriz Glenn Close en LinkedMDB, y el de la cantante Glenn Close en MusicBrainz, la consulta solo podría llevarse a cabo en caso de que el usuario conociera ambos vocabularios y las IRIs que representan a la artista en ambos, luego el enfoque explorativo se reduciría a un enfoque federado y los inconvenientes de este. Y segundo, la superposición de dominios en los conjuntos de datos, y por tanto la existencia de vocabularios distintos para definir realidades similares o equivalentes, puede ser un problema. Esto es debido a que se pueden dar por respondidas las consultas con resultados expresados en un vocabulario concreto y darse el caso de que el usuario deseara las respuesta expresadas en otros vocabularios o incluso que quisiera encontrar la misma respuesta expresada en vocabularios de diferentes orígenes. Volvamos al ejemplo de la artista Glenn Close pero considerando conjuntos de datos que presentan una superposición como DBPedia y LinkedMDB; si quisiéramos recuperar los trabajos de la actriz, consultando únicamente la DBPedia, podríamos obtener una respuesta válida. Pero quizás a ciertos usuarios les interesaría saber si en el conjunto de datos LinkedMDB existe algún otro resultado no recogido en la DBpedia. Y estos últimos resultados se perderían si en este enfoque los resultados recolectados en la DBpedia han dado por resuelto el alcance de la consulta.

En todos los casos que hemos introducido hay dos características en común: el lenguaje de consultas SPARQL, como marco de definición de la consulta y la traducción de la consulta entre los distintos conjuntos de datos, para salvar la heterogeneidad de vocabularios. La propuesta de procesamiento de consulta que se presenta en esta tesis se basa en la traducción automática de las consultas de unos conjuntos de datos a otros, adecuándolas al vocabulario de cada conjunto. Su principal ventaja y caracteristica diferenciadora es que permite al usuario abstraerse de los aspectos técnicos que dificultan la consulta y que se listan a continuación:

1. Vocabularios distintos para la definición de cada conjunto de datos.
2. Accesibles mediante puntos de acceso heterogéneos en sus protocolos.
3. Volúmenes y actualizaciones en periodicidades variantes.

4. Niveles de enlazado y vinculación con el resto de conjuntos de la nube muy diversos.

En la tabla 1.4 puede apreciarse la gran diferencia existente entre las características técnicas de la DBpedia y LinkedMDB como conjuntos de datos de ejemplo.

Se aporta una descripción detallada del concepto de traducción en el apéndice A, la sección A.3, a continuación aportamos su definición:

La traducción de la consulta desde un conjunto de datos origen a un conjunto de datos objetivo consiste en la modificación sintáctica y eventualmente semántica de la consulta siguiendo ciertas reglas de reescritura, para que pase de estar expresada en los vocabularios del conjunto de datos origen a los vocabularios del conjunto de datos objetivo.

De nuevo recurrimos al ejemplo de la actriz Glenn Close, y consideramos una consulta en la que el usuario quiere conocer las películas en las que ha trabajado como actriz, recuperando para ello, primero, la información sobre el conjunto de datos DBpedia, que es con el que se encuentra familiarizado y, posteriormente, ampliando la respuesta en el conjunto LinkedMDB. En el primer caso, lanzaríamos la siguiente consulta, escrita en el vocabulario de la DBpedia, contra el punto de acceso (`https://dbpedia.org/sparql`) que habilita este conjunto de datos:

```
SELECT ?films
WHERE {
?films
<http://dbpedia.org/ontology/starring>
<http://dbpedia.org/resource/Glenn_Close> .
}
```

Nos devuelve un total de 53 películas, pero ¿qué ocurre si para el usuario los resultados obtenidos no son suficientes y no tiene conocimientos suficientes para plasmar la consulta en otros conjuntos de datos de los que poder extraer más información?. En este caso, el proceso de consulta basado en la traducción permitiría, a partir de la consulta anterior, obtener una consulta válida en términos de nuevos conjuntos de datos. En el ejemplo el usuario estaría interesado en el conjunto de datos LinkedMDB, con punto de acceso `http://www.linkedmdb.org/snorql/`. Luego, el proceso de traducción nos devolvería la siguiente consulta.

```
SELECT ?films
WHERE {
?films
<http://data.linkedmdb.org/resource/movie/actor>
<http://data.linkedmdb.org/resource/actor/29776>  .
}
```

Esta consulta devuelve 48 resultados. Su volumen es menor que el devuelto por la DBpedia pero, al analizarlos, los conjuntos no coinciden y podemos ver que el conjunto de datos LinkedMDB aporta nuevos títulos de películas a la búsqueda del usuario. Un ejemplo de ello es el recurso `http://data.linkedmdb.org/resource/film/43183`, cuyo título es "The Chumscrubber", que en el conjunto de resultados inicial recuperado de la DBpedia no aparecía.

Tabla 1.4: Ejemplo de diferencia entre características técnicas entre conjuntos de datos DBpedia y LinkedMDB

	DBpedia	**LinkeMDB**
Vocabularios principales	Ontología DBpedia	Ontología Movie
Protocolos puntos de acceso	SPARQL 1.1 Virtuoso	SPARQL 1.0 Snorql
Volúmenes y actualizaciones	474M Triples (v 3.9)	6.148.121 Triples
Nivel de enlazado	Enlazado con más de 36 conjuntos de datos, con un total de más de 17 millones de enlaces.	Enlazado con otros 6 conjuntos de datos, con un total de 162.000 enlaces.

Hay que tener en cuenta dos puntos relevantes en todo este proceso, ya que nos permiten salvar las desventajas presentadas hasta ahora y conseguir que este enfoque sea exitoso.

- **Reglas de reescritura**: permiten llevar a cabo el proceso de traducción; para ello reescriben la consulta transformando de forma secuencial los patrones de grafo que la componen. La aplicación de las reglas está sujeta a que se cumplan ciertas precondiciones para las IRIs que aparecen en los patrones de grafo en un contexto restringido definido por la propia regla. Este contexto se extrae principalmente de los dos conjuntos de datos principales (origen y destino), un conjunto de datos (denominado "puente") que se utiliza como base de conocimiento para ayudar en la obtención de enlaces entre los conjuntos de datos origen y destino, y otros servicios y repositorios que son fuente de información sobre enlaces y alineamientos. En esta tesis se ha trabajado en la definición de cinco tipos de reglas, dependiendo de la motivación de la transformación, que son: de equivalencia, de jerarquía, basadas en las respuestas, basadas en el perfil de los recursos y basadas en las características. Además se han aportado dos formas de representación de las reglas de reescritura, una basada en la sintaxis propia de las reglas de transformación y otra basada en el lenguaje de consulta SPARQL. En el capítulo 3 puede verse una descripción introductoria de las reglas, y en las secciones A.4 y B.4 de los apéndices se puede encontrar una explicación más detallada, asociada a cada forma de representación nombrada anteriormente. En entornos reales, como la Web de datos, la existencia de pocos enlaces que forman una red de interconexiones superficial hace necesario que la definición del conjunto de reglas sea capaz de sortear esta deficiencia, aprovechando, en la medida de lo posible, el potencial de los escasos enlaces existentes y abarcando un conjunto extenso de casuísticas en las relaciones. El uso de un amplio rango de reglas permitirá obtener un conjunto mayor de traducciones para la consulta, lo que aumentará las posibilidades de obtener nuevas respuestas para el usuario. Otros aspectos relacionados con la definición de las reglas y que se han tenido en cuenta en este trabajo han sido: a) evitar la necesidad de tareas de preprocesamiento que conlleven tiempos de espera para el usuario, y b) la disponibilidad de un contexto de aplicación adecuado, que su ejecución se limite a la información disponible en la Web de datos y no dependa de otras fuentes externas, que restrinjan su utilización.

- **Modificación semántica de la consulta**. Existen múltiples técnicas como la reescritura o relajación de consulta que han sido ampliamente utilizadas en el campo de recuperación de la información, con el objetivo de aportar respuestas más amplias al usuario, a costa de relajar las condiciones de la consulta durante su ejecución modificando su semántica; ejemplos son los trabajos de [RK13], [DSW06] o [HMM13]. Pero todos estos autores se centran en relajar la consulta dentro de un único conjunto de datos. Es decir, sus técnicas de relajación no funcionarían en el escenario de esta tesis en el que se plantea que la consulta se traduzca de un conjunto de datos origen a un conjunto de datos destino, ya que se deberían tener en cuenta diferentes vocabularios. Los trabajos presentados por [MBGC12a] o [CSM+10a]si que consideran la traducción entre conjuntos de datos distintos, pero centran su investigación en la preservación semántica de la consulta durante su traducción entre diferentes conjuntos de datos pertenecientes al entorno de la nube de datos enlazados. En nuestra opinión, el que su enfoque considere solo aquellas traducciones que preserven la semántica de la consulta, es una aproximación limitada que puede producir que en muchas ocasiones no se obtenga respuesta a la consulta. En un entorno abierto y participativo como es la Web de datos, en nuestra opinión es interesante eventualmente primar la obtención de resultados frente a mantener la equivalencia semántica. Por ello en el enfoque de traducción de esta tesis optamos por no siempre conservar la semántica de la consulta, a fin de que en la mayoría de las ocasiones pueda obtenerse una respuesta aceptable y efectiva para el usuario.

En la unión de las reglas de reescritura con la posibilidad de modificar semánticamente la consulta reside una de las principales contribuciones de esta tesis, definir reglas de reescritura adecuadas para aquellos contextos en los que los conjuntos de datos no se encuentran completamente alineados, y en las que se consideren técnicas de reescritura que acepten modificaciones sintácticas o eventualmente semánticas de la consulta con el objetivo de aportar una respuesta aceptable para el usuario.

Al intentar abstraer al usuario de las complicaciones técnicas de la traducción, se corre el riesgo de que este se pierda durante al proceso, y no sea capaz de interpretar adecuadamente los resultados que se le aporten. Por ello es necesario informar al usuario de la calidad de la traducción, teniendo en cuenta las dos dimensiones que son importantes en la recuperación de información: por una parte, una medida relativa a los resultados que se entregan al usuario y por otra una medida de cómo de similar semánticamente es la traducción aportada a la consulta original, o lo que es lo mismo si se ha producido pérdida de información semántica en el proceso. Esto es esencial ya que el usuario no tiene conocimientos como para indagar en el proceso de traducción que se le ha aplicado a su consulta y por lo tanto no puede percibir o evaluar la calidad de los resultados que se le devuelvan asociados a estas traducciones, y menos en un entorno dinámico y tan poco estándar como es la Web de datos.

Para implementar ambas medidas, la calidad de los resultados y la similaridad semántica entre consultas, debemos repasar los trabajos realizados en dos campos de investigación reconocidos como son la recuperación de información IR, y el mapeo de ontologías, respectivamente.

- En el caso del primero, recuperación de información, la métrica F1 se usa a menudo en este campo para medir el rendimiento de las consultas en la obtención de resultados. El valor F1 se considera una media armónica que combina los valores de la precisión y de la exhaustividad. Para calcularla se debe obtener previamente el valor para estas métricas y para ello es necesario definir el conjunto de respuestas relevantes y el conjunto de respuestas recuperadas en el que se basa su cálculo. En nuestro trabajo, para el conjunto de respuestas relevantes hemos recurrido al conocimiento de expertos, que nos han provisto de las traducciones de las consultas en los conjuntos de datos destinos, a estas consultas las consideramos "Gold Standards". El conjunto de respuestas recuperadas son las obtenidas a partir de las traducciones devueltas por el sistema. La definición de estas métricas, así como la metodología llevada a cabo para su cálculo se encuentra descrita de forma detallada en la sección B.5. En esa parte del trabajo también presentamos cómo el problema de esta métrica reside en que es computacionalmente costosa y necesita de la intervención de un experto que provea de una "Gold Standard" en la que basar el cálculo. Todo esto hace que sea imposible su computación en el momento de ejecución de la consulta, luego necesitamos ser capaces de estimar su valor en tiempo de ejecución. Para ello, como ya hemos adelantado, se utiliza un modelo predictivo. Este modelo se basa en técnicas de aprendizaje automático, y se entrena mediante un histórico compuesto por conjuntos de consultas SPARQL y sus traducciones previstas ("Gold Standards"). De esta forma, una vez el modelo se encuentre entrenado, habrá aprendido lo suficiente como para estimar en tiempo real el valor de F1 para una consulta y conjunto de datos destino introducidos por el usuario.

- Por otra parte, la similaridad semántica se ha asociado típicamente a la relación de dos términos ontológicos. En este aspecto, el trabajo presentado en Euzenat et al. [ES13b], es un completo análisis sobre este tipo de medida de similitud, en el que las técnicas para calcular la similitud se clasifican en: basadas en nombre, en estructura, extensionales y semánticas. Pero la diferencia es que en esta tesis se busca estimar la similitud entre consultas SPARQL. Las consultas SPARQL comprenden términos que pueden ser clases, propiedades o individuos. Por lo tanto, necesitamos basarnos en técnicas básicas que aborden la similitud entre estos conceptos para desarrollar el nivel superior que sería la métrica de similitud entre consultas. Aunque esta tarea debe reconocerse como un problema diferente, con sus dificultades propias. Como se indica en [DG13], la noción apropiada de similitud de consulta depende de la aplicación en la que se utiliza. Por eso, a lo largo de la sección B.4, presentamos

cómo nuestra medida de similitud semántica se acopla al proceso de traducción y sus reglas de reescritura con el fin de ser lo más representativa posible.

En el ejemplo utilizado, como hemos visto, el número de resultados y ciertos valores de estos entre uno y otro conjunto no coincidían. Pero además la sintaxis y semántica de la consulta también han variado. La variación de sintaxis entre ambas consultas está clara ya que los términos de ambas consultas son diferentes, pero los cambios semánticos son más difíciles de detectar y han venido generados por las reglas de reescritura aplicadas. En el ejemplo, la modificación semántica se ha producido al traducir la propiedad `<http://dbpedia.org/ontology/starring>` por `<http://data.linkedmdb.org/resource/movie/actor>`, ya que no existían relaciones de equivalencia y esta traducción se ha producido gracias a otro tipo de relación detectada en el momento de aplicación de las propias reglas que producía cierta pérdida de semántica. Por ello en el ejemplo se le ofrecería al usuario un indicador de la calidad de los resultados, de valor $0,81$, calculado utilizando el modelo predictivo para el indicador F1, y una medida de la similaridad entre las consultas, de valor $0,94$, calculado usando una métrica de similaridad adaptada a las reglas de reescritura aplicadas durante la traducción. El significado exacto de estas métricas y la forma en la que se calculan sus valores es una parte primordial del trabajo de esta tesis y se explica de forma reducida en el capítulo 3 y en detalle en las secciones del apéndice B.

En la literatura existen otros trabajos que intentan resolver escenarios de consulta como el que hemos descrito a lo largo de la introducción. El que cubre un mayor número de requisitos de los comentados anteriormente: facilidad de expresión de la consulta, manejo de vocabularios heterogéneos, sin preprocesamientos previos, admisión de modificaciones semánticas en la consulta y sin restricciones en la formulación de la consulta, es el presentado por [LFMS12]. Este trabajo presenta un sistema que permite la respuesta de consultas basándose en ontologías e integrando información de muy diversos orígenes, que ofrece además una interfaz basada en palabras claves. Su principal inconveniente es que no provee de la expresividad completa del lenguaje de consulta SPARQL, es decir sus consultas sufren de limitaciones de expresividad. El trabajo realizado por Fujino et al. [FF14] ofrece la posibilidad de consultar conjuntos de datos heterogéneos y admitir respuestas imprecisas, que son las dos contribuciones novedosas y primordiales que ofrece nuestro enfoque de traducción, pero ellos necesitan tareas de preprocesamiento lo que limita la aplicabilidad de su aproximación. Una explicación detallada de las diferentes características y un estudio pormenorizado de qué trabajos las cumplen y cuales no lo hacen puede verse en el apéndice A, sección A.3.

En cuanto a los trabajos existentes relativos a la estimación de la calidad de las traducciones, un estudio de los avances en este campo se puede ver en el apéndice B, sección B.2. Este análisis revela que en la literatura no existen trabajos que hayan proporcionado una métrica de calidad para la reescritura de consultas entre diferentes conjuntos de datos. El trabajo con un enfoque más cercano es el realizado por [HPW08], pero se centra

en proveer de un ranking de las consultas relajadas mediante su modelo de reescritura y relajación, en vez de un indicador de la calidad de cada posible traducción.

1.6 Objetivo general y estructura de la tesis

La presente tesis doctoral profundiza en la siguiente cuestión: Ausencia en la Web de Datos de mecanismos de consulta eficientes capaces de abstraer al usuario de la complejidad derivada de la cantidad y heterogeneidad de los conjuntos de datos y vocabularios disponibles.

En esta tesis se presenta una posible solución que realiza una traducción entre los diferentes vocabularios. Con traducción nos referimos a transformar una consulta SPARQL adecuadamente expresada en términos de los vocabularios utilizados en un conjunto de datos de origen en otra consulta SPARQL adecuadamente expresada para un conjunto de datos objetivo que involucra diferentes vocabularios. La traducción se basa en alineaciones existentes entre términos en diferentes conjuntos de datos. Si no existen suficientes alineaciones de términos, se acepta que se produzca una pérdida semántica y se le ofrezca al usuario una aproximación de la consulta para evitar devolverle una respuesta vacía. Para la implementación se utilizan reglas de transformación y técnicas de reescritura de consultas SPARQL.

El documento se encuentra redactado siguiendo la norma de Tesis por compendio de contribuciones, que se recoge en el Artículo 7 de la normativa de Gestión de doctorado (BOPV de 27 de junio de 2013).

Los capítulos de esta tesis se han organizado de la siguiente forma:

- **Capítulo 1**. Presenta una breve descripción del escenario y problemas que cubre esta tesis. En él se intenta describir su relevancia, y el estado actual de las investigaciones relacionadas, analizando qué aspectos innovadores aporta nuestro trabajo.

- **Capítulo 2**. Establece el objetivo general y lista las hipótesis de investigación que se establecen en esta tesis y que han sido planteadas a partir de los problemas que se han descrito previamente. En este capítulo también se incluye la metodología de investigación que se ha seguido para la consecución de la tesis.

- **Capítulo 3**. Aporta un resumen de la solución propuesta al problema de traducción de consultas, junto con la métrica utilizada para estimar la calidad de la traducción.

- **Capítulo 4**. Recoge las principales contribuciones de esta tesis y las contrasta con las hipótesis de investigación planteadas en el capítulo 2. Además, se presentan las conclusiones obtenidas del trabajo realizado.

- **Apéndices A y B**. El apéndice A presenta el artículo principal desarrollado en esta tesís y centrado en la especificación de las reglas de reescritura. Y el segundo apéndice B recoge un artículo posterior,

que extiende al primero, presentando la metodología utilizada para estimar la calidad de la traducción y ser capaces de evaluar el sistema.

Capítulo 2

Objetivos y metodología de investigación

En este capítulo primero enunciamos el objetivo general que ha promovido esta tesis, junto con las diferentes hipótesis de investigación que se han planteado, considerando el entorno y los problemas anteriormente introducidos, relativos al procesamiento de consultas sobre la Web de datos. Posteriormente presentamos la metodología utilizada para intentar alcanzar los diferentes objetivos mediante la validacíon de las hipotesís establecidas. Terminamos el capítulo enumerando los principales resultados obtenidos en forma de publicaciones.

2.1 Objetivos de investigación

La actual red de enlaces entre recursos de la Web de Datos es superficial, cambiante y escasa. Esto hace necesario que los mecanismos y aproximaciones de consulta en la Web de datos se sofistiquen para sacar el máximo potencial a los existentes. Los enfoques actuales de consulta: centralizado, federado y exploratorio, no ofrecen una solución eficiente. Como hemos mencionado en la sección anterior, el enfoque centralizado tiene dos inconvenientes, su coste en integración y la necesidad de sincronizar los datos con los repositorios padres. En el caso del enfoque federado, la desventaja sigue siendo la necesidad de seguir conociendo el "vocabulario federado" al completo y el trabajo necesario para integrar en la federación cada conjunto de datos que desee incorporarse. Y por último en el caso del enfoque exploratorio, sus problemas se asocian con su dependencia a la existencia de interconexiones entre los conjuntos de datos y su alto coste computacional. Esto nos lleva a considerar otro enfoque en el que el usuario únicamente inicia el proceso de consulta sobre el conjunto de datos en el que está acostumbrado, con lo cual únicamente tiene que conocer sus vocabularios, y mediante un mecanismo de traducción iría consultando incrementalmente un conjunto de datos tras otro. Basándonos en lo anterior, el objetivo general en esta tesis es el siguiente:

Objetivo 1 *Obtener traducciones a la consulta que satisfagan con sus respuestas al usuario en un entorno tan dinámico y parcialmente vinculado como es la Web de datos.*

A continuación, listamos las diferentes hipótesis sobre las que se ha trabajado, presentando primero una explicación de las consideraciones que nos han llevado a establecer cada una de ellas.

Actualmente, optar por la preservación semántica de la traducción de la consulta como opción principal de la traducción, en el entorno que conforma la Web de datos, es una utopía. En nuestra opinión, es preferible ofrecer traducciones que, aunque no preserven la semántica de la consulta, ofrezcan una opción aceptable y efectiva, en lugar de tratar de mantener la traducción semánticamente equivalente y no ofrecer ninguna respuesta al usuario. Un enfoque rígido y restrictivo que no permita reglas de reescritura con modificación de la semántica de la consulta no es apropiado para el estado actual de la Web de datos, donde los orígenes y dominios son múltiples y los enlaces son escasos. Es decir, hay que explorar gran cantidad de datos sin que sus relaciones se encuentren definidas con una semántica clara y precisa. Esto permite al usuario valorar resultados, que no hubiera explorado siguiendo un enfoque restrictivo en cuanto a la pérdida de cierta semántica en la consulta. Todo lo anterior no quita para que la preservación semántica sea siempre la primera opción.

Hipótesis 1 *Las reglas de reescritura son una herramienta adecuada para un enfoque de consulta mediante traducción. Permitir que las reglas de reescritura transformen la consulta mediante modificaciones durante el proceso de traducción, es una opción válida para obtener resultados más amplios y, en ocasiones, la única opción para ofrecer un resultado.*

El usuario, mediante el proceso de traducción, puede conseguir abstraerse de ciertas dificultades técnicas, en especial de los diferentes vocabularios. Pero esto conlleva que en el momento en que el usuario necesite interpretar los resultados, le sea imposible establecer cómo de confiables son estos, y en qué magnitud su calidad responde a sus niveles de exigencia. Y es que no puede medirlo al no ser consciente del proceso que se ha llevado a cabo. Una opción de solucionar esta dificultad y volver mensurable la traducción consiste en aportarle un indicador, representativo y legible. Este indicador tiene que tener en cuenta las dos dimensiones involucradas, por una parte, la variación en la extensión de los resultados devueltos, y por otra en qué medida se ha modificado la semántica de la consulta durante la traducción.

Hipótesis 2 *Puede ofrecerse una métrica de calidad del proceso de traducción que tenga en cuenta tanto la calidad de las respuestas dadas como la existencia eventual de modificaciones semánticas respecto a la consulta original, para de esta forma hacer interpretable el proceso de traducción al usuario.*

Los enfoques de consulta en la Web de datos deben intentar reducir al mínimo los tiempos de espera de los usuarios al tratarse de sistemas on-line. En la literatura, la mayoría de sistemas de reescritura de consultas [RK13, DSW06, HMM13, FCPW17] o traducción entre conjuntos de datos [MBGC12b], no aportan información sobre sus tiempos de ejecución medios. En nuestro caso nos parece crucial que los tiempos de espera del

usuario sean inferiores a decenas de segundos, y que la traducción pueda ser parte del tiempo de procesamiento de la consulta, sin necesidad de pasos previos o tareas de pre-procesamiento costosas y que limiten su aplicación en la Web de datos.

Hipótesis 3 *El proceso de traducción puede lograrse con un rendimiento en tiempos que sea aceptable para la interacción con el usuario.*

El entorno de la Web de datos demanda una solución capaz de abarcar diferentes escenarios y responder a diversos usuarios, debido a su carácter participativo, y que se adapte a su evolución y naturaleza cambiante. La información de las diferentes traducciones que la solución vaya gestionando son una importante fuente de conocimiento, y por ello creemos que podría ser de gran ayuda su uso como históricos, en concreto durante el entrenamiento del modelo encargado de la estimación del valor del indicador F1.

Hipótesis 4 *Se puede aprender de la experiencia acumulada de las traducciones que vayan realizándose, con el objetivo de perfeccionar el sistema, especialmente las métricas de calidad aportadas al usuario.*

En la literatura actual no tenemos constancia de ningún marco de evaluación, ni conjunto de pruebas pensado para sistemas de reescritura de consultas SPARQL. Por lo tanto, la definición de un marco de evaluación, que permita medir hasta qué punto el sistema es útil, y que sirva para el mantenimiento y perfeccionamiento del propio sistema, como base de pruebas de rendimiento y marco de comparación con otros sistemas, nos parece una contribución valiosa a la comunidad científica. En nuestro caso, los dos factores a evaluar principalmente son la valoración de las reglas de reescritura utilizadas y la calidad de los resultados ofrecidos al usuario. Luego dentro del marco será necesario definir un conjunto de pruebas adecuado al escenario de traducción sobre conjuntos de datos parcialmente alineados, y seleccionar o diseñar métricas que representen los indicadores adecuados.

Hipótesis 5 *Se puede desarrollar un marco de evaluación y conjuntos de prueba adecuado para los sistemas de traducción sobre diferentes conjuntos de datos, ya que es una necesidad a día de hoy, que permitiría establecer comparativas justas para avanzar en este campo de investigación.*

2.2 Metodología de investigación

Para alcanzar la consecución del objetivo general anteriormente presentado y validar las hipótesis establecidas, se propuso la estrategia y fases que puede verse en la figura 2.1, y que se explica en detalle a continuación:

1. Selección de los conjuntos de datos. Al iniciar la investigación y adentrarnos en el entorno de los datos abiertos y enlazados, nos encontramos con una ingente cantidad de conjuntos de datos (en junio de 2018 la cifra era de 1,229 conjuntos de datos[1]), que nos era

[1] https://lod-cloud.net/

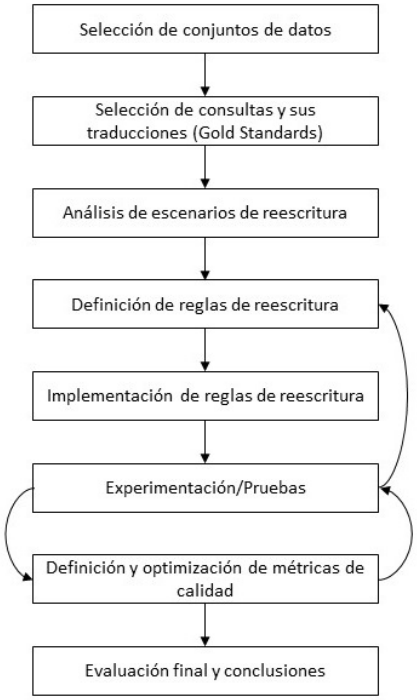


FIGURA 2.1: Estrategia y fases de la investigación realizada.

necesario analizar para determinar los problemas y retos a los que realmente nos enfrentábamos. Por lo tanto, la primera tarea a llevar a cabo era la selección de conjuntos de datos representativos y de un tamaño asumible y adecuado para su análisis. Esta selección se encuentra detallada en la sección 3.1 de la solución propuesta. Una vez analizamos los conjuntos de datos en base a sus posibilidades de consulta, dominios a los que pertenecían y calidad tanto de sus datos como vocabularios, pudimos determinar una muestra adecuada en la que basar la investigación.

2. Selección de las consultas y sus traducciones (Gold Standards). El siguiente paso se centró en recolectar una batería de consultas representativas de los diferentes tipos de consultas que los usuarios estaban interesados en realizar sobre los conjuntos de datos seleccionados en el punto anterior. Para ello se recolectaron consultas heterogéneas de distintos logs reales de puntos de acceso SPARQL y de conocidos juegos de pruebas utilizados en estudios previos, por ejemplo, QALD[2] o FedBench [SGH+11a]. El problema es que estas consultas estaban definidas en el vocabulario del conjunto de datos origen, y era necesario obtener su traducción al vocabulario del conjunto de datos destino, lo que hemos denominado Gold Standards, y de este modo conocer las necesidades y dificultades técnicas que este proceso planteaba. La Gold Standard asociada a cada consulta se obtuvo gracias a la colaboración de expertos en el dominio y vocabulario de los conjuntos de datos involucrados en la traducción. Estas dos tareas se describen en la sección 3.1 y están relacionadas con fases posteriores de evaluación, ya que las consultas tienen un papel relevante en la definición de un marco de evaluación adecuado.

3. Análisis de escenarios de reescritura. La selección del conjunto de consultas y la generación de sus traducciones (Gold Standard) nos permitió determinar las debilidades actuales existentes en el procesamiento de consultas mediante la traducción. A partir del conocimiento adquirido, fuimos capaces de establecer los escenarios básicos que debía abordar el proceso de traducción, y delimitar el alcance de la reescritura que se debería llevar a cabo sobre las consultas durante su aplicación. Durante esta fase se llevó a cabo un completo análisis del estado del arte, para buscar enfoques y propuestas cercanas a nuestro planteamiento de la consulta en el entorno de la Linked Open Data mediante un proceso de traducción entre los vocabularios de los diferentes conjuntos de datos. El resultado de este estudio puede verse en la sección A.2.

4. Definición de reglas de reescritura. Analizando las consultas, los términos que aparecían en ellas, y como estos se encontraban definidos en los diferentes conjuntos de datos, fuimos capaces de establecer: primero, que la rigurosa preservación de la semántica de la consulta no era el único enfoque válido y segundo, que para llevar a cabo

[2] https://qald.sebastianwalter.org/

la traducción era necesario fijar una serie de motivaciones y procedimientos en los que basar las reglas de reescritura. A partir de estas consideraciones, definimos cinco tipos de reglas: de equivalencia, de jerarquía, basadas en las respuestas, basadas en el perfil de los recursos y basadas en las características. El trabajo asociado a la definición de las reglas se encuentra explicado en la sección 3.2.

5. Implementación de las reglas de reescritura. Tras la definición de las reglas de reescritura, estudiamos cual sería la arquitectura conveniente para el sistema que estábamos planteando y las tecnologías más adecuadas para su implementación. La arquitectura resultante y la explicación asociada a cada uno de los módulos que la componen se puede ver en la sección 3.3. Además, llevamos a cabo la implementación de dicha arquitectura en un prototipo. Esto nos permitió establecer un enfoque iterativo de pruebas, e ir adaptando y mejorando nuestro prototipo en función de los resultados de estas.

6. Experimentación/Pruebas. Durante esta fase se desarrollaron pruebas de funcionalidad y rendimiento. El resultado de las pruebas permitió mejorar el conjunto de muestras (consultas y gold standards), las reglas de reescritura y su implementación y detectar la necesidad de especificar medidas con las que validar el comportamiento del sistema.

7. Definición y optimización de métricas de calidad. En las etapas anteriores nos centramos en la implementación del proceso de traducción, sin reflejar ni validar la calidad de los resultados que le ofrecíamos al usuario final. Por eso vimos la necesidad de una fase posterior en la que definiríamos el modo de medir la calidad del proceso de traducción y como mostrársela al usuario, evitando que el proceso de traducción se convirtiera para él en una caja negra. En esta fase se define una medida en la que se reflejará la calidad del proceso de traducción. La medida se compone de dos valores: el factor de similaridad de las consultas y la predicción de la F1 de sus resultados. Su explicación se incluye en la sección 3.2.6. La medida es adaptable dependiendo del tipo de usuario y los escenarios bajo los que se utiliza el sistema. Por ello en esta fase se planteó el diseño de dos procesos: el primero para la optimización de los parámetros de las métricas de similaridad, y el segundo para predecir el valor de F1, ya que la disponibilidad de las Gold Standards definidas por expertos no era factible en el momento de ejecución. Ambos procesos utilizan el histórico de consultas para aprender del comportamiento del sistema y configurarlo de forma adecuada.

8. Evaluación final del sistema y conclusiones. Para la evaluación final del sistema se definió un marco de evaluación adecuado y concebido para un entorno real. Este se encuentra explicado a lo largo de la sección 3.4, y principalmente se compone de una batería de consultas y métricas capaces de medir la calidad de las traducciones. Una vez evaluado el sistema, es necesario extraer conclusiones útiles a

partir del análisis de los resultados obtenidos. El análisis completo se encuentra de nuevo en la sección 3.4.

2.3 Publicaciones

Las publicaciones resultantes del conocimiento adquirido a lo largo de la investigación de esta tesis y en las que se basa el trabajo presentado en este documento han sido las siguientes:

- En revistas indexadas (se encuentran incluidas en los apéndices A y B, respectivamente):

  Ana I. Torre-Bastida, Jesús Bermúdez, and Arantza Illarramendi. A rule-based transducer for querying incompletely aligned datasets. ACM Trans. Web, 12(4):23:1-23:40, September 2018. [TBBI18b].

  Ana I Torre-Bastida, Jesús Bermúdez, and Arantza Illarramendi. Estimating query rewriting quality over lod. Semantic Web, (Preprint):1-26, 2018.[TBBI18a].

- En conferencias:

  Ana I Torre-Bastida, Jesús Bermúdez, Arantza Illarramendi, Eduardo Mena, and Marta González. Query rewriting for an incremental search in heterogeneous linked data sources. In International Conference on Flexible Query Answering Systems, pages 13-24. Springer, 2013. [TBBI+13].

  Ana I Torre-Bastida. Incremental sparql query processing. In Extended Semantic Web Conference, pages 712-716. Springer, 2013. [TB13].

  Ana I Torre-Bastida, Jesús Bermúdez, Eduardo Mena, and Arantza Illarramendi. Diseño de un repositorio rdf basado en tecnologías nosql. In JISBD 2011. XVI Jornadas de Ingeniería del Software y Bases de Datos. Springer, 2011. [TBBMI11].

Capítulo 3

Solución propuesta

En este capítulo se condensa un resumen de todo el trabajo llevado a cabo en esta tesis, siguiendo la metodología propuesta y con referencias a los apéndices A y B donde se encuentran los artículos publicados que presentan una explicación más detallada. Las primeras fases de investigación se centraron en la selección de conjuntos de datos del entorno de la Linked Open Data y juegos de consultas que resultaran relevantes para esclarecer los objetivos y el alcance de la tesis. Por lo tanto, la primera sección de este capítulo describe el trabajo realizado en estas etapas generando una batería de consultas adecuada. En la sección posterior se presenta el componente principal de todo nuestro trabajo, el mecanismo capaz de realizar la traducción de consultas por la vía de su transformación gradual en base a un conjunto de reglas de reescritura. En esta misma sección también se incluye la presentación de dichas reglas, principalmente cómo son, cómo se aplican, las representaciones utilizadas para definirlas y los diferentes tipos, cada uno con su explicación. El proceso de traducción lleva asociado el cálculo de una medida de la calidad, por eso se expone, por cada regla, las métricas de similaridad asociadas, y se finaliza la sección con la explicación del cálculo del factor de similitud. En la siguiente sección se explica el modo en que se ha planteado un marco de desarrollo para la aproximación de consulta basada en la traducción, y cómo éste se implementa en un sistema, del que especificamos su arquitectura y la utilización de tecnologías de transformación de grafos. Y en la sección final del capítulo se introduce cómo se ha llevado a cabo la evaluación del sistema, incluyendo los procesos de optimización y configuración: cómo son el cálculo de la idoneidad del factor de similaridad y la generación del modelo predictivo F1.

3.1 Selección de conjuntos de datos y consultas

Hoy en día la nube de datos enlazados es ingente, el número de conjuntos de datos aumenta cada día, al igual que los datos que se albergan en cada uno de ellos. Por ello, al comienzo de esta tesis vimos la necesidad de seleccionar un conjunto reducido de consultas, a partir del cual comenzar la investigación sobre su procesamiento en el entorno de la Linked Open Data,

para que esta tarea no resultara inabordable. El primer paso consistió en seleccionar los conjuntos de datos sobre los que trabajaríamos. En nuestro caso, decidimos que todos los conjuntos de datos debían pertenecer al entorno de los datos abiertos y enlazados, y poner puntos de acceso a disposición de los usuarios. Los conjuntos de datos se restringieron a tres dominios: audiovisual, bibliográfico y ciencias de la naturaleza, y se eligieron aquellos que presentaran mayor número de enlaces a otros datasets. En total se consideraron 17 conjuntos de datos reales, y un conjunto de datos sintético, una explicación más completa puede encontrarse en la sección B.5.1.

El siguiente paso se centró en seleccionar distintos tipos de consultas definidas en los vocabularios de los conjuntos de datos ya seleccionados. Para que las consultas representaran un entorno real, en un principio se extrajeron estas de puntos de acceso SPARQL reales (como el de la BNE o DBpedia), pero apenas había diversidad sintáctica y las variaciones en el tipo de operadores utilizados y en la estructura de las consultas era escasas. Por ello se amplió la selección de las consultas experimentales a juegos de pruebas reconocidos en la literatura como son: QALD[1] o FedBench [SGH+11a]. Pero disponer de consultas definidas en el vocabulario de un conjunto de datos, no es suficiente para analizar el proceso de traducción. Para ello necesitamos consultas iniciales definidas en el vocabulario del conjunto de datos origen, y su traducción realizada por un experto al vocabulario del conjunto de datos destino, lo que se denomina *Gold Standard*. Para determinar la Gold Standard asociada a cada consulta se preguntó a expertos en el dominio y vocabulario de los conjuntos de datos involucrados en la traducción. El proceso de la selección de consultas y la generación de las Gold Standards queda ampliamente descrito en la sección B.5.1.

3.2 Traductor basado en reglas

La traducción, como ya hemos indicado anteriormente, es el proceso mediante el cual transformamos una consulta inicial para que pase de estar expresada en los vocabularios del conjunto de datos origen a los vocabularios del conjunto de datos objetivo mediante ciertas reglas de reescritura, que la modifican sintácticamente y, eventualmente, semánticamente. Las reglas de reescritura producen una consulta semánticamente equivalente siempre y cuando se encuentren relaciones de equivalencia entre el vocabulario del conjunto de datos origen $V(D_s)$ y el vocabulario del conjunto de datos destino $V(D_t)$. Estas relaciones de equivalencia son triples del tipo "links RDF externos" en los que la propiedad representa la equivalencia y el sujeto representa un recurso del conjunto de datos origen y el objeto un recurso del conjunto de datos destino, o viceversa. Estos enlaces pueden extraerse de diferentes fuentes accesibles en la Web de datos: VoId linksets, servicios de correferencia, repositorios de mappings, etc. En el

[1] https://qald.sebastianwalter.org/

resto de casos, en que no se encuentren este tipo de enlaces, el proceso intentará conseguir, por medio de las transformaciones, una consulta similar semánticamente hablando.

Continuando con el ejemplo presentado inicialmente en el capítulo 1, vamos a extenderlo para que nos sirva como base para la explicación de las diferentes reglas de reescritura.

Consideremos una consulta en la que queremos recuperar los títulos de las películas, sus directores y los lugares de nacimiento de estos, para todas aquellas películas en las que ha participado Glenn Close como actriz y Arthur Schmidt como editor. Además, se incluye la condición de que los directores sean considerados artistas. Inicialmente el usuario estará interesado en consultar la DBpedia y posteriormente solicitará ampliar la información en el conjunto de datos LinkedMDB.

```
SELECT ?films ?director ?place
WHERE {
?films
<http://dbpedia.org/ontology/starring>
<http://dbpedia.org/resource/Glenn_Close> .
?films
<http://dbpedia.org/ontology/director>
?director .
?director a
<http://dbpedia.org/ontology/Artist>
?director
<http://dbpedia.org/ontology/birthPlace>
?place.
?films
<http://dbpedia.org/ontology/editing>
<http://dbpedia.org/resource/Arthur_Schmidt_(film_editor)> .
}
```

Las reglas permiten la reescritura de los patrones de grafo de la consulta de forma gradual. La aplicación de las reglas está sujeta a que se cumplan ciertas precondiciones para las IRIs que aparecen en los patrones de grafo en un contexto restringido definido por la propia regla, lo que hemos definido como *contexto de aplicación*. El contexto de aplicación se extrae principalmente de los dos conjuntos de datos principales (origen y destino), un conjunto de datos, que se utiliza como base de conocimiento, para ayudar en la localización de enlaces entre los conjuntos de datos origen y destino y que denominamos *conjunto de datos puente*, y otros servicios y repositorios que son fuente de información sobre enlaces y alineamientos. En el caso del ejemplo, el contexto estaría formado por los conjuntos de datos DBpedia y LinkedMDB, junto al conjunto de datos puente Freebase, y otros servicios como sameAs.org[2]. Las reglas se aplican hasta que todos los términos de los patrones de grafos se encuentran en el vocabulario del conjunto de datos destino. El algoritmo de aplicación puede verse en el apéndice A, algoritmo 1.

En nuestro trabajo hemos utilizado dos formas para la representación de las reglas, acomodándose así a distintos tipos de audiencia:

[2] `http://sameas.org/`

- La primera representación está basada en secuentes. La explicación completa de esta forma de representación puede verse en la sección A.4 del apéndice correspondiente, junto con la descripción de las reglas en esta sintaxis. En resumen, siendo Q la consulta a ser traducida, y C el contexto de la consulta en el que se unen los tres datasets involucrados: source Ds, target Dt y bridge Db y las alineaciones entre sus terminos. La regla de reescritura

$$\frac{CG}{Q[QP] \Longrightarrow Q[RP]}$$

indica cómo al darse la circunstancia de que el grafo de contexto de la propia regla CG se satisface en el contexto C de la consulta, se puede producir la reescritura, que consiste en reemplazar todas las ocurrencias del patrón QP en la consulta Q, por el patrón RP.

A continuación visualizamos un ejemplo de la regla de equivalencia en su variante E2 en esta forma de representación:

$$\frac{(u, eq, t:u_i)(i = 1..n)}{Q[(u, p, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}(t:u_i, p, o)]} \tag{3.1}$$

- La segunda representación se basa en la sintaxis del lenguaje de consultas SPARQL. La adaptación a las consultas SPARQL está descrita en la sección B.3 del apéndice correspondiente, y es una variación de la sentencia CONSTRUCT en SPARQL.

```
REPLACE   template
BY        template
WHEN {
    graph pattern
}
```

En ella se definen tres nuevas palabras reservadas: *REPLACE*, *BY* y *WHEN*, las dos últimas se asemejan a las cláusulas *CONSTRUCT y WHERE* respectivamente. La palabra *template* representa un patrón de grafo descrito en las especificaciones de las cláusulas REPLACE y BY. A continuación, explicamos cómo las reglas se plasman en este formato: REPLACE especifica una plantilla, *template*, que debe coincidir con una parte del patrón de grafo definido en la consulta que va a ser reescrita. Se corresponde con la parte [QP] en la anterior forma de representación. Esta coincidencia es el desencadenante de la regla. Dicha parte coincidente en la consulta a ser reescrita será reemplazada por la plantilla, *template*, especificada en la cláusula BY (se corresponde la parte [RP] en la anterior representación). La cláusula WHEN permite especificar una serie de patrones de grafo que deben satisfacerse en el contexto formado por los conjuntos de datos para que la reescritura se produzca. En la representación previa se trataba de la parte CG. Como hemos comentado, se asemeja a la cláusula WHERE y si no se satisface no se aplicará la regla.

A continuación volvemos a mostrar un ejemplo de la regla de equivalencia en su variante E2 en esta forma de representación:

```
PREFIX s:<source dataset>
PREFIX t:<target dataset>
PREFIX b:<bridge dataset>
REPLACE  s:u #p #o .
BY       UNION(t:u #p #o)
WHEN {
    s:u ?eq1 b:u .
    b:u ?eq2 t:u .
    FILTER (?eq1 = owl:sameAs ||
            ?eq1 = owl:equivalentClass ||
            ?eq1 = owl:equivalentProperty ||
            ?eq1 = skos:exactMatch)
    FILTER (?eq2 = owl:sameAs ||
            ?eq2 = owl:equivalentClass ||
            ?eq2 = owl:equivalentProperty ||
            ?eq2 = skos:exactMatch)
}
```

En esta tesis se ha trabajado en la definición de cinco tipos de reglas, dependiendo de la motivación de la transformación, que son: de equivalencia, de jerarquía, basadas en las respuestas, basadas en el perfil de los recursos y basadas en las características. Otro de los puntos relevantes a considerar en la explicación de estas reglas es la posible modificación de la semántica de la consulta. Esto, como ya hemos introducido en secciones anteriores, conlleva la necesidad de proveer al usuario con una medida de la similaridad, denominada factor de similaridad SF, que se asocia a la secuencia de aplicación de las reglas de reescrituras para la consulta. Por eso cada regla utiliza una métrica de similaridad adecuada a su naturaleza. En las secciones posteriores se explican de forma resumida cada una de las clases de reglas, y cuáles son las medidas de similaridad asociadas. Una explicación más detallada puede encontrarse en la sección B.4.

3.2.1 Reglas de equivalencia

El objetivo de este tipo de reglas es transformar la consulta en otra equivalente en el conjunto de datos destino. Las relaciones de equivalencia que se tienen en cuenta son dos:

1. Triples de equivalencia, formados por los predicados: $owl:sameAs$, $owl:equivalentClass$, $owl:equivalentProperty$, $skos:exactMatch$.
2. Alineamientos expresados en lenguaje EDOAL[3].

Si al analizar la consulta existen términos que no se encuentran en el vocabulario del conjunto de datos destino, y estos se encuentran vinculados en el contexto mediante alguna de las relaciones anteriores a términos del conjunto de datos destino, se produce la aplicación de esta regla.

Del contexto es de donde se extraen los diferentes mapeos de equivalencia considerados en el primer tipo de regla de equivalencia listado. Existen siete variantes de las reglas de equivalencia, dependiendo de la posición del término a traducir en el patrón de grafo (sujeto, predicado u

[3] http://alignapi.gforge.inria.fr/edoal.html

objeto) y de la utilización o no del conjunto de datos *bridge*, además de la variante que utiliza los alineamientos expresados en lenguaje EDOAL. Su representación en forma de reglas se encuentra detallada en la figura 3.1.

$$(E1)\ \frac{EDOAL : ELHS \rightarrow t{:}ERHS}{Q[ELHS] \Longrightarrow Q[ERHS]}$$

$$(E2)\ \frac{(u, eq, t{:}u_i)(i = 1..n)}{Q[(u, p, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}(t{:}u_i, p, o)]}$$

$$(E3)\ \frac{(u, eq, t{:}u_i)(i = 1..n)}{Q[(s, u, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}(s, t{:}u_i, o)]}$$

$$(E4)\ \frac{(u, eq, t{:}u_i)(i = 1..n)}{Q[(s, p, u)] \Longrightarrow Q[\text{UNION}_{i=1..n}(s, p, t{:}u_i)]}$$

$$(E5)\ \frac{(u, eq, b{:}u_i)(i = 1..n)(b{:}u_i, eq, t{:}u_{ij})(j = 1..n_i)}{Q[(u, p, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}^{j=1..n_i}(t{:}u_{ij}, p, o)]}$$

$$(E6)\ \frac{(u, eq, b{:}u_i)(i = 1..n)(b{:}u_i, eq, t{:}u_{ij})(j = 1..n_i)}{Q[(s, u, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}^{j=1..n_i}(s, t{:}u_{ij}, o)]}$$

$$(E7)\ \frac{(u, eq, b{:}u_i)(i = 1..n)(b{:}u_i, eq, t{:}u_{ij})(j = 1..n_i)}{Q[(s, p, u)] \Longrightarrow Q[\text{UNION}_{i=1..n}^{j=1..n_i}(s, p, t{:}u_{ij})]}$$

FIGURA 3.1: Reglas de equivalencia.

A continuación, presentamos la definición de la regla de equivalencia E5 en la forma de representación basada en SPARQL, es la variante de esta regla que reemplaza el sujeto de un patrón de triple. La definición completa de este tipo de reglas según esta representación puede verse en la sección B.4.

```
PREFIX s:<source dataset>
PREFIX t:<target dataset>
PREFIX b:<bridge dataset>
REPLACE  s:u  #p  #o .
BY       UNION(t:u #p  #o)
WHEN {
    s:u  ?eq1  b:u .
    b:u ?eq2  t:u  .
    FILTER (?eq1 = owl:sameAs ||
            ?eq1 = owl:equivalentClass ||
            ?eq1 = owl:equivalentProperty ||
            ?eq1 = skos:exactMatch)
    FILTER (?eq2 = owl:sameAs ||
            ?eq2 = owl:equivalentClass ||
            ?eq2 = owl:equivalentProperty ||
            ?eq2 = skos:exactMatch)
}
```

En el ejemplo, las reglas de equivalencia serían las primeras en aplicarse y una parte del contexto de aplicación para esta regla sería el siguiente:

```
<http://dbpedia.org/ontology/director> owl:sameAs
<http://data.linkedmdb.org/resource/movie/director>.
<http://dbpedia.org/resource/Glenn_Close> owl:sameAs
<http://data.linkedmdb.org/resource/actor/29776>.
<http://dbpedia.org/ontology/editing> owl:sameAs
<http://data.linkedmdb.org/resource/movie/editor>.
```

Basándonos en los triples del contexto, se aplicaría la regla E3 sobre la propiedad *director* y la propiedad *editing* y la regla E4 sobre el recurso que identifica a la actriz Glenn Close, quedando la consulta intermedia de la siguiente forma:

```
SELECT ?films ?director ?place
WHERE {
?films
<http://dbpedia.org/ontology/starring>
 <http://data.linkedmdb.org/resource/actor/29776> .
?films
 <http://data.linkedmdb.org/resource/movie/director>
?director .
?director a
<http://dbpedia.org/ontology/Artist> .
?director
<http://dbpedia.org/ontology/birthPlace>
?place.
?films
 <http://data.linkedmdb.org/resource/movie/editor>
<http://dbpedia.org/resource/Arthur_Schmidt_(film_editor)> .
}
```

Hay que tener en cuenta que, al tratarse de relaciones de equivalencia, no se ha producido pérdida de semántica durante el proceso, por lo que la similaridad de las reglas de equivalencia es 1, $\phi(u) = 1$.

3.2.2 Reglas de jerarquía

El motivo de esta clase de reglas es el de transformar la consulta por medio de la generalización o especialización de sus términos.

Los predicados de alineamiento que se consideran pueden ser: $skos : narrower$, $skos : broader$, $rdfs : subClassOf$, $rdfs : subPropertyOf$. Y dependiendo del alineamiento utilizado pueden darse dos situaciones de generalización o especialización, según ocurra lo siguiente, respectivamente:

1. Un término en el patrón de triple se encuentra enlazado con una colección de términos más generales definidos en el vocabulario del conjunto de datos destino, y por lo tanto se reescribe generalizando la semántica de la consulta.

2. Un término en el patrón de triple se encuentra enlazado con una colección de términos más específicos definidos en el vocabulario del conjunto de datos destino, y por lo tanto se reescribe especializando la semántica de la consulta.

Existen seis variantes de este tipo de regla que se encuentran detalladas en la figura 3.2, dependiendo de la posición del término a traducir en el patrón de grafo (sujeto, predicado u objeto) y dependiendo de si el alineamiento expresa generalización o especialización del término, ya que la reescritura se produce mediante el operador AND, intersección de los terminos más generales, o UNION, unión de los terminos más especificos, respectivamente. Además, en cada variante se consideran dos versiones dependiendo de si la relación de jerarquía esta expresada en un solo alineamiento, son aquellas reglas númeradas con el sufijo *a*, o en dos niveles definidos en alineamientos distintos, que son las reglas numeradas con el sufijo *b*.

Un ejemplo de la definición de una regla de jerarquía en la forma de representación basada en SPARQL puede verse a continuación. Se muestra la variante H9a de esta regla, que reemplaza el predicado de un patrón de triple con un solo nivel de alíneamiento. Esta representación se encuentra explicada de forma más detallada en la sección B.4.

```
PREFIX s:<source dataset>
PREFIX t:<target dataset>
REPLACE  #s  s:p  #o .
BY       AND(#s t:p  #o)
WHEN {
    s:p  ?sub  t:p .
    FILTER (?sub = rdfs:subPropertyOf ||
            ?sub = skos:narrower)
}
```

Las reglas de jerarquía se aplican después de haberse aplicado las reglas de equivalencia y quedar aún términos sin traducir. En el ejemplo, en el contexto de aplicación se ha encontrado la siguiente relación jerárquica:

```
<http://dbpedia.org/ontology/Artist> rdfs:subClassOf
<http://xmlns.com/foaf/0.1/Person> .
```

Por lo que se aplicaría la regla H9a sobre la clase Artist, quedando la consulta intermedia de la siguiente forma:

```
SELECT ?films ?director ?place
WHERE {
?films
<http://dbpedia.org/ontology/starring>
 <http://data.linkedmdb.org/resource/actor/29776> .
?films
 <http://data.linkedmdb.org/resource/movie/director>
?director .
?director a
 <http://xmlns.com/foaf/0.1/Person> .
?director
<http://dbpedia.org/ontology/birthPlace>
?place.
?films
 <http://data.linkedmdb.org/resource/movie/editor>
<http://dbpedia.org/resource/Arthur_Schmidt_(film_editor)> .
}
```

En este caso, se ha generalizado la consulta, produciéndose cierta pérdida en la semántica de la consulta de la que se deberá informar al usuario.

$$(H8a)\quad \frac{(u, sub, t:u_i)(i = 1..n)}{Q[(u, p, o)] \Longrightarrow Q[\text{AND}_{i=1..n}(t:u_i, p, o)]}$$

$$(H8b)\quad \frac{(u, sub, v_i)(i = 1..n)(v_i, sub, t:u_{ij})(j = 1..n_i)}{Q[(u, p, o)] \Longrightarrow Q[\text{AND}_{i=1..n}^{j=1..n_i}(t:u_{ij}, p, o)]}$$

$$(H9a)\quad \frac{(u, sub, t:u_i)(i = 1..n)}{Q[(s, u, o)] \Longrightarrow Q[\text{AND}_{i=1..n}(s, t:u_i, o)]}$$

$$(H9b)\quad \frac{(u, sub, v_i)(i = 1..n)(v_i, sub, t:u_{ij})(j = 1..n_i)}{Q[(s, u, o)] \Longrightarrow Q[\text{AND}_{i=1..n}^{j=1..n_i}(s, t:u_{ij}, o)]}$$

$$(H10a)\quad \frac{(u, sub, t:u_i)(i = 1..n)}{Q[(s, p, u)] \Longrightarrow Q[\text{AND}_{i=1..n}(s, p, t:u_i)]}$$

$$(H10b)\quad \frac{(u, sub, v_i)(i = 1..n)(v_i, sub, t:u_{ij})(j = 1..n_i)}{Q[(s, p, u)] \Longrightarrow Q[\text{AND}_{i=1..n}^{j=1..n_i}(s, p, t:u_{ij})]}$$

$$(H11a)\quad \frac{(t:u_i, sub, u)(i = 1..n)}{Q[(u, p, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}(t:u_i, p, o)]}$$

$$(H11b)\quad \frac{(v_i, sub, u)(i = 1..n)(t:v_{ij}, sub, v_i)(j = 1..n_i)}{Q[(u, p, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}^{j=1..n_i}(t:v_{ij}, p, o)]}$$

$$(H12a)\quad \frac{(t:u_i, sub, u)(i = 1..n)}{Q[(s, u, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}(s, t:u_i, o)]}$$

$$(H12b)\quad \frac{(v_i, sub, u)(i = 1..n)(t:v_{ij}, sub, v_i)(j = 1..n_i)}{Q[(s, u, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}^{j=1..n_i}(s, t:v_{ij}, o)]}$$

$$(H13a)\quad \frac{(t:u_i, sub, u)(i = 1..n)}{Q[(s, p, u)] \Longrightarrow Q[\text{UNION}_{i=1..n}(s, p, t:u_i)]}$$

$$(H13b)\quad \frac{(v_i, sub, u)(i = 1..n)(t:v_{ij}, sub, v_i)(j = 1..n_i)}{Q[(s, p, u)] \Longrightarrow Q[\text{UNION}_{i=1..n}^{j=1..n_i}(s, p, t:v_{ij})]}$$

FIGURA 3.2: Reglas de jerarquía.

La medida de similaridad asociada es una adaptación de la propuesta en [ZZLY02] y se puede ver su explicación completa en la sección B.4 del apéndice correspondiente. En esta aplicación de la regla, la función de similaridad entre la clase Artist y la clase Person resulta ser: $\phi(dbo : Artist) = 0.6$.

3.2.3 Reglas basadas en la respuesta

El motivo de esta clase de reglas es utilizar los recursos que se consideran respuestas a la pregunta en el conjunto de datos origen como ejemplos de cuáles son los recursos objetivo en el conjunto de datos destino. Una vez que hemos detectado estos *recursos objetivo*, a partir de ellos recreamos el patrón de triple necesario en el conjunto de datos destino. La aplicación de este tipo de regla está restringida a aquellos triples en los que el resto de elementos del triple ya se encuentran expresados usando el vocabulario del conjunto de datos destino o son variables.

Existen seis variantes de este tipo de reglas que se encuentran detalladas en la tabla 3.3, varían en función del rol del término a traducir: sujeto, predicado u objeto, y del rol de los términos que ya se encuentren traducidos dentro del triple considerado. Las dos últimas variantes $A18$ y $A19$ se aplican sobre patrones del estilo $(?x, ?p, u)$ y $(u, ?p, ?x)$, en los que puede verse que no hay términos que ya se encuentren traducidos. Y se utiliza la función $mostFrequent()$ para obtener el término que más frecuentemente ha aparecido entre los triples del conjunto de datos destino relacionados y que describen al término a traducir que se pasa como parámetro a dicha función, y que será el que finalmente reemplace en la consulta al término a traducir. Una definición más detallada y formal de esta función puede verse en la sección A.4.3, junto a la descripción completa de esta clase de reglas en base a esta representación.

De nuevo este tipo de reglas también puede formalizarse siguiendo una sintaxis de lenguaje de consultas SPARQL, esta representación se encuentra descrita en la sección B.4. A continuación, vemos el caso correspondiente para la regla A19 basada en respuestas que traduce el sujeto:

```
PREFIX s:<source dataset>
PREFIX t:<target dataset>
REPLACE s:u ?p ?x
BY      t:u ?p ?x
WHEN {
    ?s as t:u (COUNT(?s) as ?OCCURNUM)
    WHERE {
        s:u ?p ?x .
        ?x ?eq t:?o .
        ?s t:?pt t:?o .
        FILTER (?eq = owl:sameAs ||
                ?eq = skos:exactMatch)
    }
    GROUP BY ?s ORDER BY DESC ?OCCURNUM
    LIMIT 1
}
```

Las reglas basadas en respuesta se aplican cuando las reglas de equivalencia y jerarquía ya han sido aplicadas y se dan las condiciones de

$$(A14)\quad \frac{\begin{array}{c}Answers(?x, t{:}p, u) = [x_1, \ldots, x_n] \\ (i = 1..n)(x_i, eq, t{:}x_{ij})(j = 1..n_i)(t{:}x_{ij}, t{:}p, t{:}o_{ij}^k)(k = 1..n_{ij})\end{array}}{Q[(?x, t{:}p, u)] \Longrightarrow Q[\text{UNION}_{i=1..n}(\text{AND}_{j=1..n_i}^{k=1..n_{ij}}(?x, t{:}p, t{:}o_{ij}^k))]}$$

$$(A15)\quad \frac{\begin{array}{c}Answers(u, t{:}p, ?x) = [x_1, \ldots, x_n] \\ (i = 1..n)(x_i, eq, t{:}x_{ij})(j = 1..n_i)(t{:}s_{ij}^k, t{:}p, t{:}x_{ij})(k = 1..n_{ij})\end{array}}{Q[(u, t{:}p, ?x)] \Longrightarrow Q[\text{UNION}_{i=1..n}(\text{AND}_{j=1..n_i}^{k=1..n_{ij}}(t{:}s_{ij}^k, t{:}p, ?x))]}$$

$$(A16)\quad \frac{\begin{array}{c}Anwers(?x, u, t{:}o) = [x_1, \ldots, x_n] \\ (i = 1..n)(x_i, eq, t{:}x_{ij})(j = 1..n_i)(t{:}x_{ij}, t{:}p_k, t{:}o)(k = 1..r)\end{array}}{Q[(?x, u, t{:}o)] \Longrightarrow Q[\text{AND}_{k=1..r}(?x, t{:}p_k, t{:}o)]}$$

$$(A17)\quad \frac{\begin{array}{c}Anwers(t{:}s, u, ?x) = [x_1, \ldots, x_n] \\ (i = 1..n)(x_i, eq, t{:}x_{ij})(j = 1..n_i)(t{:}s, t{:}p_k, t{:}x_{ij})(k = 1..r)\end{array}}{Q[(t{:}s, u, ?x)] \Longrightarrow Q[\text{AND}_{k=1..r}(t{:}s, t{:}p_k, ?x)]}$$

$$(A18)\quad \frac{mostFrequent(?x, ?p, u) = [t{:}o]}{Q[(?x, ?p, u)] \Longrightarrow Q[(?x, ?p, t{:}o)]}$$

$$(A19)\quad \frac{mostFrequent((u, ?p, ?x)) = [t{:}s]}{Q[(u, ?p, ?x)] \Longrightarrow Q[(t{:}s, ?p, ?x)]}$$

FIGURA 3.3: Reglas basadas en la respuesta.

restricción anteriormente explicadas. Siguiendo con el ejemplo anterior, vemos que el único patrón sobre el que se pueden aplicar estas reglas es el siguiente:

```
?films <http://data.linkedmdb.org/resource/movie/editor>
<http://dbpedia.org/resource/Arthur_Schmidt_(film_editor)> .
```

Imaginemos que una respuesta en el conjunto de datos origen es:

```
<http://dbpedia.org/resource/Primary_Colors_\%28film\%29>
```

Y contamos con el siguiente contexto de aplicación:

```
<http://dbpedia.org/resource/Primary_Colors_\%28film\%29> owl:sameAs
<http://data.linkedmdb.org/resource/film/15706>.
<http://data.linkedmdb.org/resource/film/15706>
<http://data.linkedmdb.org/resource/movie/editor>
<http://data.linkedmdb.org/resource/editor/1815>.
```

Por lo tanto, de aquí inferimos que el recurso `<http://dbpedia.org/resource/Arthur_Schmidt_(film_editor)>` puede ser reemplazado por `<http://data.linkedmdb.org/resource/editor/1815>`. Se ha simplificado el número de respuestas y el contexto de aplicación para evitar restar

claridad al ejemplo, pero la realidad es que en esta clase de reglas el número de triples que componen el contexto suele ser muy numeroso por lo que su aplicación suele ser costosa computacionalmente. Un ejemplo más detallado puede apreciarse en el apéndice A. La consulta final, resultante de la aplicación de la regla, sería de la siguiente forma:

```
SELECT ?films ?director ?place
WHERE {
?films
<http://dbpedia.org/ontology/starring>
 <http://data.linkedmdb.org/resource/actor/29776> .
?films
 <http://data.linkedmdb.org/resource/movie/director>
?director .
?director a
 <http://xmlns.com/foaf/0.1/Person> .
?director
<http://dbpedia.org/ontology/birthPlace>
?place.
?films
 <http://data.linkedmdb.org/resource/movie/editor>
 <http://data.linkedmdb.org/resource/editor/1815> .
}
```

Gracias a esta regla se han encontrado resultados que de otra forma no se habrían considerado, y para ello no nos hemos basado en alineamientos que aseguren una relación de este tipo:

```
<http://dbpedia.org/resource/Arthur_Schmidt_(film_editor)> owl:sameAs
<http://data.linkedmdb.org/resource/editor/1815>
```

Este hecho indica que no podemos asegurar la preservación semántica de la consulta, por lo que es necesario calcular el grado de modificación semántica que la consulta ha sufrido durante la aplicación de la regla. La métrica de similaridad encargada de esta tarea se define como la combinación lineal de otras tres:

$$\begin{aligned} S(u,v) &= \alpha_n \cdot S_n(u,v) + \alpha_d \cdot S_d(u,v) + \alpha_o \cdot S_o(u,v) \\ & \alpha_n, \alpha_d, \alpha_o \geq 0 \quad \wedge \quad \alpha_n + \alpha_d + \alpha_o = 1 \end{aligned}$$

Donde S_n y S_d son métricas basadas en métodos de cadenas de texto, y S_o es la distancia jerárquica anteriormente utilizada para calcular la similaridad en las reglas de jerarquía y que se encuentra basada en la propuesta de Zhong et al.[ZZLY02]. S_n es la media aritmética de las distancias de Levenshtein y Jaccard y S_d tiene en cuenta el contexto de los términos, construyendo un modelo de *Bag of words* con los triples del perfil de cada uno y finalmente calculando la distancia *Cosine* entre los vectores construidos mediante ese modelo. Además los parámetros α_n, α_o y α_d son valores que nos permiten ponderar este conjunto de métricas según se quiera asignar un peso mayor a unas u otras dependiendo del entorno o caso de uso en el que estemos. Para optimizar el valor de estos parámetros proponemos un proceso dedicado que aprenda del uso dado al sistema, y

que se explica en detalle en la sección B.5.2. El valor de similaridad para el término traducido mediante la aplicación de esta regla es el siguiente: $\phi(db-r:Arthur\_Schmidt\_(film\_editor)) = 0.81$.

3.2.4 Reglas basadas en el perfil

El motivo de estas reglas consiste en utilizar todos aquellos triples que describen al recurso en el conjunto de datos origen, es decir aquellos triples en los que el recurso actúa como sujeto, predicado u objeto. La heurística considera que, si alguno de los recursos del perfil tiene una relación de equivalencia en el conjunto de datos destino, y se descubre que en el perfil del recurso destino existe un recurso lo suficientemente similar con el recurso a traducir, este último será el nuevo término adecuado. Para ello se necesita una función de similaridad que nos indique en qué medida el recurso es susceptible de ser el adecuado.

Existen tres variantes para este tipo de regla que se encuentran detalladas en la tabla 3.4, dependiendo del rol del término a traducir: sujeto, predicado u objeto. Un aspecto muy importante a tener en cuenta en esta clase de reglas es el uso de la función de similaridad para reducir el contexto de aplicación y seleccionar el término final al que se traduce la consulta. La función *maxSim* es la encargada de esto, para ello, se establece un umbral, h, que sirve para descartar todos aquellos términos, w, con valores de similitud que no superen su valor, ya que se consideran insuficientemente similares al término a traducir. De los diferentes términos que pasan el filtro, la función elige el que presente un mayor valor de similitud. Por defecto, en nuestros experimentos utilizamos un valor de umbral de $h = 0.45$ para equilibrar la cantidad de IRI descartadas y aceptadas, este valor se ha seleccionado a partir de un estudio realizado con juegos de pruebas preparados al efecto. La explicación detallada puede verse en la sección A.4.4.

$$(P20)\quad \frac{(i=1..n)(a_i \in \mathcal{V}_{D_s}(u))(a_i, eq, t{:}a_{ij})(j=1..n_i) \quad R = \cup_{i=1..n}^{j=1..n_i} \mathcal{V}_{D_t}(t{:}a_{ij}),\ \ t{:}ms = maxSim(u,h,R)}{Q[(u,p,o)] \Longrightarrow Q[(t{:}ms,p,o)]}$$

$$(P21)\quad \frac{(i=1..n)(a_i \in \mathcal{V}_{D_s}(u))(a_i, eq, t{:}a_{ij})(j=1..n_i) \quad R = \cup_{i=1..n}^{j=1..n_i} \mathcal{V}_{D_t}(t{:}a_{ij}),\ \ t{:}ms = maxSim(u,h,R)}{Q[(s,p,u)] \Longrightarrow Q[(s,p,t{:}ms)]}$$

$$(P22)\quad \frac{(i=1..n)(a_i \in \mathcal{E}_{D_s}(u))(a_i, eq, t{:}a_{ij})(j=1..n_i) \quad R = \cup_{i=1..n}^{j=1..n_i} \mathcal{F}_{D_t}(t{:}a_{ij}),\ \ t{:}ms = maxSim(u,h,R)}{Q[(s,u,o)] \Longrightarrow Q[(s,t{:}ms,o)]}$$

FIGURA 3.4: Reglas basadas en el perfil.

A continuación se muestra la regla basada en el perfil P22 encargada de la traducción de la propiedad, siguiendo la sintaxis del lenguaje de consultas SPARQL, tal y como se explica en la sección B.4.

```
REPLACE  #s s:p  #o .
BY       #s t:p  #o .
WHEN {
    ?s s:p ?o .
    ?s ?eq ?ts .
    ?o ?eq ?to .
    {?ts t:p ?to .}
    UNION
    {?to t:p ?ts .}
    FILTER (?eq = owl:sameAs ||
            ?eq = skos:exactMatch)
    FILTER (t:p = maxSim(s:p, h, profile(s:p))
}
```

Siguiendo con nuestro ejemplo existen dos términos aún sin traducir en la consulta:<http://dbpedia.org/ontology/birthPlace> y <http://dbpedia.org/ontology/starring>. En el caso del primero su perfil no ha devuelto ningún recurso que muestre relaciones de equivalencia con términos del conjunto de datos destino. Con el segundo recurso sí se han encontrado este tipo de relaciones, formándose el siguiente contexto de aplicación para la regla (por abreviar solo hemos incluido una parte de los triples que lo formarían):

```
db-r:Orlando_Bloom db-o:starring
db-r:Pirates_of_the_Caribbean:_The_Curse_of_the_Black_Pearl .
movie:actor/29718 owl:SameAs db-r:Orlando_Bloom .
movie:film/755 movie:actor movie:actor/29718 .
movie:actor/29718 movie:actor-name "Orlando Bloom" .
db-r:Johnny_Depp db-o:starring
db-r:Pirates_of_the_Caribbean:_The_Curse_of_the_Black_Pearl .
movie:actor/30028 owl:SameAs db-r:Johnny_Depp .
movie:film/755 movie:actor movie:actor/30028 .
```

A partir de este contexto se forman las parejas de transformaciones a las que se le aplica la función de similaridad, que es la misma que se ha utilizado en las reglas basadas en la respuesta. Con los valores obtenidos conseguimos un ranking ordenado de las posibles reescrituras para el término:

$$\begin{aligned} \phi(\texttt{movie:actor}, \texttt{db-o:starring}) &= 0.934 \\ \phi(\texttt{movie:actor-name}, \texttt{db-o:starring}) &= 0.861 \end{aligned}$$

Entre las opciones anteriores la regla selecciona la de mayor valor de similaridad, sustituyendo el predicado *db-o:starring* por *movie:actor*, quedando la consulta de la siguiente forma:

```
SELECT ?films ?director ?place
WHERE {
?films
 <http://data.linkedmdb.org/resource/movie/actor>
 <http://data.linkedmdb.org/resource/actor/29776> .
?films
```

$$(F23)\ \frac{(u, s{:}p_i, s{:}o_i)(i = 1..n)}{Q[(u,p,o)] \Longrightarrow Q[(?v,p,o)\text{AND}_{i=1..n}(?v, s{:}p_i, s{:}o_i)\ ?v\ \textit{a new variable}]}$$

$$(F24)\ \frac{(s{:}s_i, u, s{:}o_i)(i = 1..n)}{Q[(s,u,o)] \Longrightarrow Q[\text{AND}_{i=1..n}(s{:}s_i, ?v, s{:}o_i)\ ?v\ \textit{a new variable}]}$$

$$(F25)\ \frac{(s{:}s_i, s{:}p_i, u)(i = 1..n)}{Q[(s,p,u)] \Longrightarrow Q[(s,p,?v)\text{AND}_{i=1..n}(s{:}s_i, s{:}p_i, ?v)\ ?v\ \textit{a new variable}]}$$

FIGURA 3.5: Reglas basadas en las características

```
 <http://data.linkedmdb.org/resource/movie/director>
?director .
?director a
 <http://xmlns.com/foaf/0.1/Person> .
?director
<http://dbpedia.org/ontology/birthPlace>
?place.
?films
 <http://data.linkedmdb.org/resource/movie/editor>
 <http://data.linkedmdb.org/resource/editor/1815> .
}
```

De nuevo se ha producido una pérdida semántica. La medida de similaridad asociada es $\phi(movie : actor) = 0.934$, que ya ha sido calculada utilizando las mismas métricas que en las reglas basadas en respuestas.

3.2.5 Reglas basadas en las características

Se trata de la última opción para traducir un término, ya que todas las reglas anteriores han sido aplicadas y descartadas. En este caso la modificación de la semántica de la consulta es alta, ya que el término se reemplaza por una variable, a la que se le añaden una serie de condiciones en la consulta. Estas condiciones se construyen a partir del término a reemplazar, y se basan en los triples en los que el término actuaba de sujeto en el conjunto de datos origen. Una vez estos nuevos patrones de triple se han incluido en la consulta con el término a reemplazar sustituido por una variable, se vuelven a aplicar las reglas de equivalencia y jerarquía. Y si después de esta segunda oportunidad las nuevas condiciones no se encuentran en el vocabulario del conjunto de datos destino, se eliminan.

Las variantes para este tipo de regla se encuentran detalladas en la tabla 3.5. Son tres, dependiendo del rol del término a traducir: sujeto, predicado u objeto.

La expresión de la regla basada en características siguiendo la sintaxis basada en lenguaje SPARQL puede verse a continuación para la variante F25 que traduce el término que actúa como objeto. Una explicación más detallada de esta representación se encuentra en la sección B.4.

```
PREFIX s:<source dataset>
PREFIX t:<target dataset>
REPLACE #s #p s:u
BY      #s #p ?v .
        AND(?v #f #o)
WHEN {
      s:u #f #o  }
```

En el ejemplo sigue quedando un término por sustituir `<http://dbpedia.org/ontology/birthPlace>`, al que se le aplica la regla de características. El contexto se forma con los triples en los que actúa como sujeto esta propiedad, devolviendo pares (predicado, objeto) como los siguientes (entre muchos otros que por espacio no incluimos):

```
(rdf:type, rdf:Property)
(rdfs:comment, "where the person was born (en)")
(rdfs:range, dbo:Place)
(rdfs:domain, dbo:Person)
```

Estos pares se añaden a la consulta ligados a la variable por la que se ha sustituido el término, y se aplican de nuevo las reglas de equivalencia y jerarquía. Al no existir relaciones de este tipo para los nuevos términos, las reglas no pueden aplicarse, y los triples previamente añadidos se deben eliminar. Luego, finalmente, la IRI ha sido sustituida por una variable $?p$. La pérdida semántica en este caso es completa, la consulta se ha generalizado totalmente para ese patrón de triple, quedando la consulta final en:

```
SELECT ?films ?director ?place
WHERE {
?films
 <http://data.linkedmdb.org/resource/movie/actor>
 <http://data.linkedmdb.org/resource/actor/29776> .
?films
 <http://data.linkedmdb.org/resource/movie/director>
?director .
?director a
 <http://xmlns.com/foaf/0.1/Person> .
?director
 ?p
?place.
?films
 <http://data.linkedmdb.org/resource/movie/editor>
 <http://data.linkedmdb.org/resource/editor/1815> .
}
```

En el conjunto de resultados final es relevante destacar que en un principio no se encontró ningun resultado en la consulta original, pero que al traducirla y relajar las condiciones de la consulta, debido a la generalización de la regla de jerarquía y la eliminación de la condición de lugares de nacimiento de los directores, ha sido posible encontrar un resultado para la película "Sunset Boulevard", con el IRI `http://data.linkedmdb.org/resource/film/222`.

3.2.6 Medida de calidad de la traducción

La medida de la calidad de la traducción, como ya hemos indicado anteriormente, se basa en dos métricas complementarias: el factor de similaridad de las consultas y la predicción de la F1 de los resultados.

El *factor de similaridad* asociado a la consulta traducida es la agregación de los valores de similaridad devueltos por cada una de las reglas aplicadas durante el proceso de traducción. Para la agregación nos hemos basado en la distancia Euclídea, donde el punto y sus coordenadas se corresponden conceptualmente con la consulta y los términos que no se encuentran definidos en los vocabularios del conjunto de datos destino, llamémosles *términos no adecuados*. El valor asociado al término es la medida de similaridad, siendo el peor valor posible el 0, y el mejor el 1. La fórmula es la siguiente:

$$\mathcal{SF}((u_i)_{i=1}^N) = 1 - \frac{1}{\sqrt{N}}\sqrt{\sum_{i=1}^{N}(1-\phi(u_i))^2}$$

Donde $(u_i)_{i=1}^N$ representa los términos, que concuerdan con la representación de la secuencia de reglas aplicadas $(r_i)_{i=1}^N$. La explicación de esta métrica esta detallada a lo largo de la sección B.4. En el ejemplo, la consulta final ofrece un factor de similaridad $SF = 0,78$.

La medida que se ha elegido para informar sobre la calidad de los resultados es la *F1*. Esta medida combina los valores de la *precision* y de la *recall*. En el campo de estudio de la "recuperación de información", la *precision* es la fracción de instancias relevantes entre las instancias recuperadas, mientras que la *recall* (también conocida como sensibilidad o exhaustividad) es la fracción de instancias relevantes que se han recuperado sobre el total de instancias relevantes. Tanto la *precision* como la *recall* se basan, por lo tanto, en una comprensión y medida de la relevancia de los resultados. Para calcular la F1 es necesario definir:

- El conjunto de respuestas relevantes: En nuestro caso, las respuestas relevantes se asocian con la Gold Standard. En esta tesis, la Gold Standard es una consulta generada a partir de conocimiento experto sobre el conjunto de datos destino, por ello para construirlas hemos recurrido a expertos, que nos han proporcionado las traducciones de las consultas en los vocabularios de los conjuntos de datos destino.

- El conjunto de respuestas recuperadas: Las respuestas recuperadas se identifican con las respuestas obtenidas a partir de la consulta traducida por el sistema.

Gracias a estos dos conjuntos somos capaces de calcular la *precision*, *recall* y F1. Todo este proceso se encuentra ampliamente detallado en la sección B.5. El problema reside en que, en el momento de la consulta, no es factible disponer de la Gold Standard, ya que el experto debe construirla expresamente, por eso confiamos en métodos predictivos que, a partir de una base de conocimiento formada por consultas ya traducidas y las F1

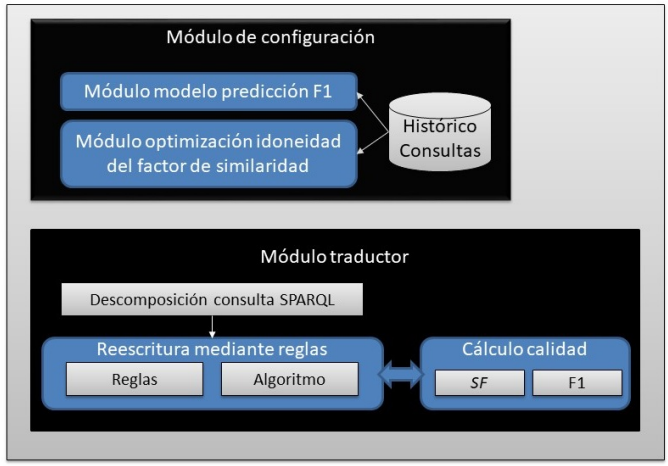


FIGURA 3.6: Visión conceptual de la arquitectura del sistema

ya calculadas, predigan el valor de F1 para nuevas consultas. Después de experimentar con varios algoritmos y varias selecciones de características para generar el modelo, el seleccionado fue *RandomForest*. Todo este proceso se muestra en la sección B.5.4. En el ejemplo, para la consulta final, se obtiene un valor de $F1 = 0,76$.

3.3 Implementación del sistema y prototipo

El proceso de consulta basado en la traducción puede plantearse como un marco de desarrollo, compuesto por diferentes elementos:

- Reglas de reescritura.
- Algoritmo de traducción.
- Mecanismo de cálculo de la similaridad de la traducción.
- Mecanismo de predicción de la calidad de los resultados.

La explicación detallada de este marco se encuentra en la sección B.3. El objetivo final es que éste se materialice en un sistema siguiendo una implementación concreta que es la que presentamos a continuación y que se encuentra detallada en las secciones B.4 y A.4. El aspecto principal a destacar de este sistema es la utilización de tecnologías de transformación de grafos para implementar las reglas de reescritura.

En la figura 3.6 puede verse una visión conceptual de la arquitectura final del sistema. El sistema se compone de dos módulos principales: el módulo traductor y el módulo encargado de la configuración del sistema.

El primero, *módulo traductor*, es el módulo central encargado del proceso de traducción de las consultas SPARQL, que se estructura a su vez

en dos componentes con los siguientes objetivos: la *reescritura mediante reglas*, y el *cálculo de la calidad*. La reescritura mediante reglas utiliza dos elementos: las *reglas* y el *algoritmo* de aplicación. Inicialmente, la consulta de origen se descompone en grafos formados por patrones de triples, y se construye el contexto de aplicación para la consulta. En este punto es cuando, siguiendo el flujo marcado por el algoritmo, las reglas comienzan a actuar mediante técnicas de transformación de grafos. Esto significa que cada regla se compone de dos partes, derecha e izquierda, cada una de ellas representada en forma de grafo. En el momento en el que alguno de los grafos o subgrafos presentes en la consulta coincidan con el de la parte de la izquierda de la regla, se sustituirá por el grafo establecido en la parte derecha. Al finalizar la ejecución de cada regla se llama al componente encargado del cálculo de la calidad, para que, en cada caso, compute la similaridad resultante de la aplicación de dicha regla. Cuando el algoritmo haya finalizado o el grafo completo haya sido transformado, la consulta vuelve a ser formulada a sintaxis SPARQL y se vuelve a utilizar el componente del cálculo de la calidad, para que, por una parte, agregue los valores de similaridad calculados en las distintas reglas aplicadas en el factor de similaridad SF y, por la otra, prediga el valor para dicha consulta de la métrica F1. De esta forma se obtiene la consulta adecuada final, con sus valores de calidad asociados $(SF, F1)$.

El segundo, *módulo de configuración del sistema*, es un módulo que se utiliza en las tareas de configuración del sistema y consta de dos componentes: *generación del modelo de predicción F1*, es el encargado de generar el modelo de aprendizaje automático que permite predecir el valor de F1 para nuevas consultas y *optimización de idoneidad del factor de similitud*, refiriéndonos a la tarea de establecer inicialmente cuales son los mejores valores posibles para los parámetros α_n, α_d y α_o que ponderan algunas de las métricas de similaridad. Ambos módulos necesitan de una base de conocimiento formada por consultas SPARQL(consulta original, traducción del sistema y Gold Standard) y su valor para la métrica F1, lo que en la figura 3.6 se denomina *histórico de consultas*.

El sistema presentado es flexible a la integración de nuevas configuraciones de reescritura (reglas y algoritmo), nuevos tipos de medidas de calidad (parámetros de similitud y modelos ML), nuevos conjuntos de datos (URL de puntos finales SPARQL) y nuevos repositorios de mapeos (URL de puntos finales SPARQL) o servicios de correferencias (API de servicio), siguiendo un modelo basado en componentes. Agregar un componente es tan fácil como proporcionar nuevos archivos de configuración en los que se describen los valores y las referencias al componente correspondiente. El sistema dispone de un prototipo implementado en lenguaje JAVA mediante el framework AGG de transformación de grafos, AGG framework[4]. Una versión de demostración se encuentra disponible online a través de la siguiente dirección: `http://sml.tecnalia.com/SparqlRewriting/faces/demoUser.xhtml`.

[4] http://www.user.tu-berlin.de/o.runge/agg/

3.4 Marco de evaluación

El marco de desarrollo y el sistema que lo implementan están concebidos en todo momento para ser evaluados en un contexto real, por lo tanto, hay una serie de elementos necesarios para llevar a cabo esta tarea: principalmente, una batería de consultas, y métricas capaces de medir la calidad de las traducciones.

Para la generación de la batería de consultas, investigamos en la literatura si existían marcos de evaluación previos en la tarea de reescribir una consulta en base a vocabularios distintos al original de la consulta, pero nos encontramos con que las pruebas eran mínimas (máximo de 5 consultas) y no podían adaptarse a nuestro sistema. Por lo tanto, fue necesario construir un juego de pruebas. Las consultas fueron extraidas de la selección y generación previa llevada a cabo en las primeras fases de nuestra investigación, que se encuentran detalladas en la sección 3.1. En total se consideraron 100 consultas para el juego de pruebas. Una vez elegidas se computaron las medidas típicas IR (del campo de investigación Recuperación de la información), como son la *precision*, *recall*, e indicador *F1*. Sus fórmulas están definidas en la sección A.5.2, el proceso completo se encuentra detallado en las secciones A.5 y B.5.1, y el apéndice con el conjunto de consultas completo se encuentra publicado en [TB18b, TB18a].

Estas consultas no solo se ven involucradas en la evaluación y análisis de los resultados, sino que también permiten mejorar el sistema, al usarlas como base de conocimiento. En nuestro trabajo estas consultas asisten en la gestión y mejora de dos procesos clave para presentarle una métrica de calidad al usuario final:

- La optimización de los parámetros de las métricas de similaridad, lo que hemos denominado *Idoneidad del factor de similaridad* y que se detalla en la sección B.5.2. Resumiendo, este proceso utiliza un algoritmo de optimización para conseguir los mejores valores para los parámetros α_n, α_d y α_o incluidos en las métricas de similaridad que se computan durante la ejecución de las reglas. La función objetivo es la maximización del número de consultas en el que la diferencia absoluta entre el valor del factor de similaridad SF y el de la medida F1 para el conjunto de consultas en estudio, sea menor que un límite establecido β.

- La generación del *modelo predictivo para F1*, debido a que, como ya hemos explicado anteriormente, las consultas Gold Standard en un escenario real no se encuentran disponibles. En este caso, el conjunto de datos del que aprende el modelo son las 100 consultas que se han comentado anteriormente, y de las que se extraen las características con las que este se entrena. Se han considerado 21 características, dividas en tres tipos: relativas a reglas y similaridad, relativas a estructura de las consultas y relativas a los conjuntos de datos en los que se ejecutan las consultas. Para conseguir el mejor modelo se han realizado pruebas con tres tipos de algoritmos y diferentes combinaciones de características, usando como métrica

de evaluación el promedio de error cuadrático, denominado R2. En todo este proceso ha resultado ganador el modelo Random Forest para el conjunto de características denotado como F1, con un valor de métrica $R2 = 0,8219$. Un mayor detalle de cómo se ha llevado a cabo la generación y evaluación del modelo predictivo se describe en la sección B.5.4.

Una vez establecido cómo hemos seleccionado nuestra batería de consultas y las métricas de evaluación, pasamos a resumir los análisis y experimentos que hemos llevado a cabo. Los puntos relevantes en que hemos centrado el análisis han sido:

- Calidad de las traducciones obtenidas por el proceso de traducción. Nuestras principales cuestiones a investigar en este punto eran dos: a) la relevancia de las diferentes clases de reglas, demostrando que ninguna de ellas era prescindible, y b) comparar las traducciones contra conocimiento experto aportado en forma de *Gold Standards*. En la primera cuestión hemos sido capaces de determinar que todas las reglas aportan valor al proceso de traducción mediante un análisis de la distribución de las consultas en base a las reglas aplicadas en cada traducción. Este análisis se encuentra presente en la sección A.5.2 del apéndice correspondiente, y muestra cómo todas las clases de reglas han sido cruciales en algún momento para conseguir alguna de las traducciones de los experimentos. En la segunda cuestión hemos comparado las traducciones y consultas Gold Standards en base a su texto y sus resultados, obteniendo que un 53% de las consultas son iguales tanto en texto como en resultados, y en un 64% no se aprecia pérdida de información en los resultados a pesar de que el texto haya variado. Solo un 4% de las consultas presentan pérdidas tanto en *precision* como en *recall*. Todo este análisis puede verse en el apéndice, sección A.5.2.

- Validación de los procesos de optimización y configuración, como son el cálculo de la idoneidad del factor de similaridad y la generación del modelo predictivo F1. Para esta tarea se realizaron una serie de experimentos con el objetivo de conseguir el mejor modelo en ambos casos, basándonos en la métrica *Error Medio Absoluto - MAE* en el caso de la optimización y del coeficiente *Promedio de error cuadrático - R2* en el caso de la predicción. Los diferentes valores registrados durante las pruebas pueden verse en la sección B.5.

- Rendimiento en tiempos. Los tiempos totales invertidos en la traducción variaron durante las pruebas desde un límite inferior de 122 ms hasta un límite superior de 11787 ms. Sin embargo, si eliminamos tres consultas que presentaban tiempos elevados, se puede concluir que cada consulta se transformó en menos de 6749 ms. El análisis de tiempos y su rendimiento esta descrito en detalle en la sección A.5.3.

Capítulo 4

Conclusiones

La Web de Datos se compone de un número cada vez mayor de conjuntos de datos, de naturaleza heterogénea y altamente distribuidos, complicando su explotación y especialmente la consulta por parte de usuarios inexpertos. En esta tesis se ha detallado una innovadora propuesta para el procesamiento inteligente de consultas sobre la Web de datos, específicamente la LODC(Linked Open Data Cloud). Esta solución consigue su objetivo mediante dos vías:

- Ofreciendo al usuario la posibilidad de explorar un mayor número de conjuntos de datos mediante la traducción de la consulta entre sus vocabularios utilizando un conjunto de reglas de reescritura. Las reglas de reescritura que componen el proceso de traducción pueden eventualmente modificar semánticamente la consulta en caso de que exista la posibilidad de que el usuario en caso contrario no consiga respuestas.

- Asistiendo al usuario en la interpretación de la calidad de los resultados, gracias a la provisión de una medida de la calidad, que mide tanto la similaridad de la consulta traducida como la calidad de las respuestas obtenidas.

La propuesta se materializa mediante la implementación de un sistema capaz de cumplir los dos puntos anteriores: desarrollando un conjunto de reglas de reescritura convenientes para el entorno de conjuntos de datos parcialmente enlazados y diseñando las métricas adecuadas para esas reglas. El sistema ha sido evaluado en un contexto real por medio de un prototipo y un marco de evaluación, ambos diseñados y configurados al efecto. Analizando los resultados obtenidos se concluye de nuestro sistema lo siguiente:

- Respecto a su propósito y características principales: Nuestro sistema tiene entre sus características el permitir el uso de respuestas imprecisas, es decir permite la modificación semántica de la consulta, en un entorno de conjuntos de datos distribuidos y parcialmente enlazados, sin centralizar previamente los repositorios.

- Respecto al rendimiento: las métricas de rendimiento en tiempos no son aportadas por la mayoría de sistemas presentes en la literatura y

dedicados a la tarea de reescritura de consultas, por lo que a pesar de no poder asegurar que nuestro sistema sea el más rápido, si hemos demostrado que sus tiempos de respuesta no superan el orden de segundos, lo cuál es aceptable para la interacción con el usuario. En cuanto a la calidad de sus resultados, el porcentaje de consultas traducidas con una similaridad semántica superior al 75% es del 90% lo que muestra una capacidad de ofrecer respuestas aceptables muy alta.

4.1 Contribuciones y resultados

En esta sección presentamos las principales contribuciones de esta tesis, mientras las evaluamos brevemente contra las hipótesis de investigación propuestas en el capítulo 2. A continuación, se listan de forma resumida y posteriormente se detalla cada contribución junto con los resultados obtenidos.

- Definición de una nueva propuesta de consulta, basada en la traducción de ésta de un conjunto de datos a otro de forma incremental.
- Definición de un conjunto de reglas de reescritura y del mecanismo de traducción.
- Definición de métricas de calidad: Factor de similitud y predicción de F1.
- Evaluación del sistema, ofreciendo resultados aceptables y que superan al estado del arte actual.
- Implementación y prototipo del sistema, configurable e implementado mediante técnicas de transformación de grafos.

4.1.1 Enfoque de traducción de consultas como nueva aproximación de consulta de la Web de datos

La Web de datos presenta unas características determinadas que dificultan su consulta, como son la heterogeneidad en la forma de modelar los datos dependiendo de su origen y la declaración ambigua de entidades entre los diferentes orígenes. Ante esto ya comentamos durante la introducción la necesidad de establecer un nuevo enfoque de consulta, y creemos que la utilización del proceso de traducción de forma incremental entre los diferentes conjuntos de datos es una aproximación útil y aceptable para el usuario. A lo largo de todo este documento hemos ido estableciendo cómo se ha conseguido su implementación y los diferentes resultados asociados que demuestran este enfoque (véase el capítulo 3 y los apéndices A y B). Además en el análisis de trabajos relacionados en la sección A.2 se evidencia que no existen otras aproximaciones que respondan a los requisitos establecidos de forma completa, como es el caso de nuestra solución, luego este estudio y el diseño del enfoque constituyen la primera contribución de nuestro trabajo.

4.1.2 Mecanismo de traducción y reglas de reescritura

El proceso de traducción mediante reglas de reescritura es nuestra contribución principal, y permite la adecuación de una consulta SPARQL de un conjunto de datos origen a otro de destino con vocabularios distintos. La principal aportación reside en la modificación semántica de la consulta por parte de ciertas reglas de reescritura, en las ocasiones en que mantener la semántica equivalente supone el riesgo de no aportar respuestas al usuario. En la sección 3.4 se ha resumido la evaluación de la solución propuesta y, en especial, el análisis realizado en cuanto a la calidad de las traducciones. En él, una de las conclusiones de la discusión ha sido cómo a pesar de no haber sido estrictos en la preservación semántica de la consulta se ha conseguido que un 64% de las consultas hayan sido traducidas sin que el usuario sufra de una disminución de la calidad de los resultados. Todo esto nos permite demostrar que la modificación semántica es vital para nuestra solución y debe ser una aproximación aceptada en los enfoques de reescritura sobre la Web de datos, quedando validada la hipótesis 1.

El conjunto de reglas de reescritura definido y el estudio de la motivación adecuada para cada tipo ha sido otro de nuestros aciertos. Otra contribución ha sido demostrar que los tipos de reglas considerados y diseñados en nuestra solución cubren las diferentes casuísticas de consulta sobre los datos abiertos enlazados, y ninguno de ellos es innecesario. Todo esto se ve reflejado en la sección 3.4, donde analizamos que todas las reglas son relevantes, ya que todas las clases de reglas han sido cruciales en algún momento para conseguir alguna de las traducciones de los experimentos. Este análisis nos permite dar por comprobada la hipótesis 1.

4.1.3 Definición de métricas de calidad: Factor de similitud y predicción de F1

El proceso de traducción es opaco para el usuario, por lo tanto, hay que ofrecerle métricas de calidad que le faciliten interpretar los resultados que recibe: la consulta traducida y sus respuestas sobre el conjunto de datos destino. Nuestra aportación es ofrecer, como parte de los resultados al usuario, una métrica de calidad compuesta por un factor de la similitud de la consulta traducida y un valor de predicción de F1. La definición y cálculo de estas dos métricas, que se basan en medidas ampliamente reconocidas, cuya interpretación es sencilla, y están adaptadas y son representativas del proceso de traducción, es otra de nuestras contribuciones que responde a la hipótesis 2, validada al aportar estas métricas junto al resultado.

En el caso del factor de similaridad es representativo del proceso ya que se calcula adaptado a las reglas de reescritura utilizadas. Además, los parámetros de las funciones de similaridad utilizadas para su cálculo se optimizan en base al conocimiento extraído de un histórico de consultas, lo que favorece su adaptabilidad al entorno de consulta. El análisis y resultados de este proceso de optimización puede verse en la sección 3.4 y más en detalle en el apéndice B.

En el caso de la predicción de F1, la métrica en la que se basa es ampliamente conocida, pero no puede ser calculada en tiempo de consulta ya que en ese momento es imposible disponer de la Gold Standard, por eso es necesario predecirla. Por ello analizamos la utilización de diferentes algoritmos seleccionando diferentes conjuntos de características hasta dar con el adecuado para construir el modelo predictivo, todo esto se encuentra descrito en la sección 3.4 y más en detalle en el apéndice B. El ganador es el algoritmo Random Forest con el conjunto completo de características ya que ofrece los mejores resultados, con un valor $R2 > 0,82$.

4.1.4 Evaluación del sistema

Respecto a los conjuntos de pruebas utilizados para la evaluación, hemos analizado la literatura existente para encontrar marcos de evaluación válidos para nuestro sistema. Y al no tener constancia de su existencia, creemos que es una de nuestras principales contribuciones el haber generado una batería de pruebas propia formada a partir de conjuntos de datos representativos y seleccionando las consultas de juegos de referencia ampliamente conocidos como son: QALD[1], FedBench [SGH+11a], y puntos de acceso SPARQL reales (como el de la BNE o DBpedia). Esto se corresponde con los requisitos establecidos en un principio en la hipótesis 5.

Entre las hipótesis presentadas, se planteaba el conseguir resultados aceptables en tiempos, es la hipótesis 3 que consideramos como primordial. Y como hemos presentado en la sección 3.4, donde se ofrece un resumen del análisis y la evaluación del sistema, podemos considerar validada la hipótesis ya que el rendimiento en tiempos del proceso de traducción se encuentra entre $122ms$ y $11787ms$, siendo el tiempo medio menor de 6749 ms, lo que es válido para un sistema de consulta on-line. En cuanto a la calidad de los resultados ofrecidos a los usuarios se puede establecer que en un 64% de los casos los resultados relevantes para el usuario son iguales a los resultados ofrecidos a este. Por lo que, de nuevo, se vuelve a confirmar que el requisito presentado en el objetivo general 1 de ofrecer traducciones adecuadas al usuario se ha cumplido.

4.1.5 Implementación y prototipo del sistema

La implementación del sistema, capaz de resolver el procesamiento de consultas en la Web de datos mediante la aproximación de traducción incremental de consultas presentado, es nuestra contribución final. Esta implementación y su prototipo se presentan en la sección 3.3 y se caracteriza por ser: configurable, ya que permite la personalización de ciertos de sus componentes, como el conjunto de reglas, el algoritmo de ejecución o las métricas de similaridad utilizadas, e inteligente, las métricas de calidad se configuran mediante procesos de optimización o predicción que pueden ir aprendiendo según el sistema recoge nuevas experiencias de consulta de

[1] https://qald.sebastianwalter.org/

los diferentes usuarios y escenarios. El hecho de que el sistema sea "inteligente", para que aprenda y responda al mayor número de casos de uso posible era uno de nuestros objetivos iniciales, hipótesis 4, que se considera validada gracias a esta contribución.

4.2 Trabajos a futuro

En esta sección tratamos dos posibles direcciones a considerar, primero las líneas basadas en posibles perfeccionamientos del trabajo actual y a continuación, trabajos a futuro que podrían servir para complementar y mejorar el sistema global.

Como líneas de perfeccionamiento de la investigación actual se pueden considerar, la inclusión en el sistema un front-end que incorporara capacidades para la consulta facetada y/o la integración de consultas en lenguaje natural. Y otras mejoras, como la búsqueda de un mayor alcance del sistema de procesamiento de consulta actual extendiéndolo con la incorporación de nuevas reglas de reescritura y medidas de similaridad, que otorgue una mayor inteligencia al sistema. Y la extensión de las pruebas actuales a un mayor número de dominios, incorporando nuevos conjuntos de datos y un mayor número de consultas, es decir ampliar el marco de evaluación actual.

En cuanto a trabajos a futuro, pensamos que las líneas de investigación más interesantes se encuentran orientadas a mejorar el tiempo de ejecución de las consultas SPARQL sobre sistemas que utilicen la federación de consultas propuesta en SPARQL 1.1. Creemos que resultaría efectivo desarrollar un sistema de planificación y paralelización de la ejecución de las consultas sobre los diferentes conjuntos de datos, para mejorar su tiempo de respuesta, disponibilidad y tolerancia en caso de fallo. En su diseño tendríamos en cuenta las características de los diferentes conjuntos de datos y sus puntos de acceso para ser capaces de evaluar distintos planes de ejecución y determinar cuál es el plan óptimo. La principal característica diferenciadora de este planificador de ejecución de consultas sería el utilizar la reescritura de las consultas para construir los diferentes planes de ejecución posibles.

Appendix A

A rule-based transducer for incompletely aligned datasets

A growing number of Linked Open Data sources (from diverse provenance and about different domains) that can be freely browsed and searched to find and extract useful information have been made available. However, access to them is difficult for different reasons. This study addressed access issues concerning heterogeneity. It is common for datasets to describe the same or overlapping domains while using different vocabularies. Our study presents a transducer that transforms a SPARQL query suitably expressed in terms of the vocabularies used in a source dataset into another SPARQL query suitably expressed for a target dataset involving different vocabularies. The transformation is based on existing alignments between terms in different datasets. Whenever the transducer is unable to produce a semantically equivalent query because of the scarcity of term alignments, the transducer produces a semantic approximation of the query to avoid returning the empty answer to the user. Transformation across datasets is achieved through the management of a wide range of transformation rules. The feasibility of our proposal has been validated with a prototype implementation that processes queries that appear in well known benchmarks and SPARQL endpoint logs. Results of the experiments show that the system is quite effective achieving adequate transformations.

A.1 Introduction

People are witnessing an explosion of types, availability, and volume of data sources accessible on the Web. In particular, the so called Web of Data is considered among the major global repositories in which the number of available linked datasets continuously increases and mainly promoted by initiatives, such as Linked Open Data, Open Government, and Linked Life Data. One main objective of these initiatives is to give access to data silos and publish their contents in a semi-structured format along with links between related data entities. As a result, a growing number of Linked Open Data sources (from diverse provenance and about different

domains) that can be freely browsed and searched to find and extract useful information have been made available.

However, access to these sources is difficult for users, with the difficulties mainly related to the highly distributed structure and evolving nature of the environment. Aspects related to volume (the number of datasets is large and their existence is difficult to verify), dynamism (datasets evolve quickly and are added and removed over time), and heterogeneity (datasets vary in size, there is no standard for source descriptions, and access options vary)[1] present specific research challenges. In this study, we mainly focused on issues concerning data heterogeneity. It is common for several datasets to describe the same or overlapped domains (e.g., Linked GeoData[2] and Geo Linked Data[3] in the geographic domain) and to use different vocabularies to describe similar information.

In this study, we describe a transducer that transforms a SPARQL query suitably expressed in terms of the vocabularies used in a source dataset into another SPARQL query suitably expressed for a target dataset involving different vocabularies. The transformation was based on existing alignments between terms in different datasets. Whenever the transducer was unable to produce a semantically equivalent query because of the scarcity of term alignments, the transducer produced a semantic approximation of the query to avoid returning an empty answer to the user. This type of transducer is useful in different scenarios, including those where an ordinary user poses a keyword-based query to a question-answering (QA) system, which constructs a SPARQL query to be run on a source dataset and then demands additional answers from a dataset with a different vocabulary (more on this in Section A.2). Another scenario might involve a scientist formulating an exploration query over a source dataset with which they are familiar and demanding additional answers by accessing different datasets while not requiring strict query equivalence, but allowing potential serendipity (notice that the scientist need not be aware of the internal structure/vocabulary of the new target dataset). A third scenario might involve an application programmer attempting to query the English DBpedia using terms extracted from a user-defined Spanish Wikipedia infobox (language type in Wikipedia notwithstanding), which is not mapped using official DBpedia terms. To obtain answers, a transformation of the source query is needed to allow adequate expression for the English DBpedia.

The contributions of this study are the transducer that goes beyond equivalence-preserving transformation, the selected set of transformation rules, and an evaluation of a prototype implementation of the transducer. The transducer transforms a SPARQL query, which is formulated using a selected source vocabulary, into another SPARQL query formulated using a target vocabulary. The transducer attempts to faithfully translate the formulated query; however, sometimes, due to a mismatch of dataset vocabularies or an incomplete definition of alignment axioms, it is impossible

[1] This resembles the terms volume, velocity, and variety used in Big Data nomenclature.

[2] http://linkedgeodata.org/

[3] http://geo.linkeddata.es/

to guarantee a transformation that preserves the semantics of the original query. In this case, the transducer offers a non-semantics-preserving transformation, but one that represents an effective approximation that is better than not providing any translation at all, because that transformation can be used by the user to obtain additional answers.

Transformation across datasets is achieved by the transducer through the management of a selected range of transformation rules. Apart from rules based on classical mappings defined between datasets (synonyms, hyponyms, hypernyms), the transducer also deals with Expressive and Declarative Ontology Alignment Language (EDOAL) [DESTdS11] alignment rules and other heuristic rules that conform to a carefully controlled set of cases (answer-based rules, profile-based rules, and feature-based rules). This allows for a greater range of query transformations, thereby increasing the chance of obtaining new answers. Rules are applied during the query processing task and accounting for locally stored mapping information and alignment-provider web services.

The proposal was validated using a prototype implementation that processed queries from a constructed benchmark. To the best of our knowledge, no benchmark has been created for the type of transformations that we are attempting. Moreover, the benchmark requires goal queries (gold standard) for comparison with the queries obtained by the transducer. Notice that such gold standards need to be designed by humans; therefore, we constructed a benchmark of 50 queries (see Section A.5) and asked human experts to design gold-standard queries (using the target-dataset vocabulary) for these selected queries. The results of the validation process are promising and presented in Section A.5.

This paper is organized as follows. In Section A.2, we present previous work related to our proposal for querying distributed and heterogeneous datasets. In Section A.3, we introduce basic concepts and notation used throughout the paper. In Section A.4, we explain the main features of the query rewriting transducer, as well as the rules that it handles. The experimental results are described in Section A.5, and we conclude with a summary of our findings.

A.2 Related work

The problem of query processing over linked data sources has been considered from different perspectives, including the development of facilities that simplify query formulation and architectural issues. Among works that facilitated query formulation, PowerAqua [LFMS12] is an ontology based QA system that offers a keyword-based query interface and is capable of locating and integrating information that can be massively distributed across heterogeneous data resources in order to return answers. AutoSPARQL [LB11] uses a supervised machine-learning technique based on positive and negative examples of query results to generate a SPARQL query. The goal of SINA [SMNA14] is to convert keywords or natural-language expressions to a SPARQL query, and the SWIP system [GTHP13] is based on the use of query patterns specific to the grounding dataset and

complex correspondences for query pattern rewriting. Query Med [SS10] allows users to query multiple biomedical-data sources by providing keywords as input, and SPARQLByE [DAB16] uses reverse engineering along with other techniques, such as query relaxation, to guide users unfamiliar with both the dataset and SPARQL to the desired query and result sets. A review by [DLSM17] presents an overview of the techniques used in current QA systems over knowledge bases, and [HWM+17] identifies common challenges, structured solutions, and provides recommendations for future QA systems.

The transducer presented in this study could complement these systems, which synthesize a SPARQL query from keywords, natural-language expressions, or query patterns. The synthetic query intends to capture the intention of the user; therefore, strict semantics-preserving transformations should not be a strong requirement. Notice that the SPARQL query is the starting point of our proposal.

With respect to the underlying architecture, two broad classes of approaches can be distinguished: *centralized repositories*, where several datasets are collected in advance, preprocessed, and stored in centralized repositories, with the queries evaluated against these centralized repositories; and *distributed* query processing, where queries are evaluated against the distributed sources. Regarding the distributed approach, two more alternatives can be distinguished: *federated* query processing (e.g., [SHH+11b, QL08b, FF14]), in which a query against a federation of data sources is split into queries that can be answered by the individual data sources; and *exploratory* query processing (e.g. [Har11]), which takes advantage of the dereferenceable internationalized resource identifiers (IRI) principle promoted by Linked Data. In the latter approach, query execution begins in a source dataset and is intertwined with the traversal of the HTTP dereferenceable IRIs to retrieve additional data from different nodes that incorporate data for answering parts of the query and include other IRIs that can be successively dereferenced to augment the queried dataset until the initial query is sufficiently answered. An interesting variation in the exploratory case is the *index-based* approach (e.g., [LT10, TUY11]) that ignores the existence of links during the query execution process and relies upon a pre-populated index that is used to identify IRIs that need to be identified during query execution. If we compare our approach with these methods, it appears more flexible. On the one hand, our approach does not require preprocessing of data sources, which is computationally costly, or the management of synchronization techniques, as in the case of centralized repositories. Additionally, it does not require the computationally costly task of building a federation. Using our approach, the user formulates a query over a source dataset with which she/he is familiar, and the system offers the possibility of enriching the obtained answer by transparently accessing additional data sources (incrementally, one data source at a time) and without being aware of the internal structure of the data sources. Moreover, our approach deals with non-preselected datasets involving heterogeneous vocabularies during the navigation process. Only a service for accessing desired datasets and companion mapping datasets

is necessary. Our approach also manages a wide range of query transformation rules (not only equivalence-type mappings, as opposed to many of the existing approaches), although it does not deal with weighted ontology mappings, which differs from a previous study [FF14].

Considering other orthogonal perspectives and regarding semantics-preserving query translation, [MBGC12b] provides a generic method for SPARQL query rewriting with respect to a set of predefined mappings between ontology schemas. Additionally, [CSM+10b] is the most similar to our approach, as it runs the same query over different datasets, although that method is more concerned with expressiveness- and alignment-model-associated limitations. However, the semantics-preserving issue is too difficult in many situations and results in failure on the part of these systems to produce any answer to the user. Our proposal addresses a scenario where a translation that preserves semantics is not a strong requirement and, therefore, it obtains semantics-preserving and not-semantics-preserving transformations to increase the opportunities of obtaining answers.

Additionally, [RK13, DSW06, HMM13, FCPW17] considered the relaxation of constraints in the query. In [RK13], the authors promote query relaxation on the fly by approximating answers through relaxing the query conditions during query execution. In [DSW06], relaxation was controlled by domain- and user-preference-ontology conditions, and in [HMM13], the authors proposed a simple knowledge organization system-based term expansion and scoring technique to improve retrieval effectiveness. In [FCPW17], the authors presented an extension of SPARQL 1.1 (SPARQL^{AR}) that incorporates query approximation and relaxation operators through regular path queries. However, all these approaches considered only a fixed-source dataset for the relaxed query with no change in vocabulary considered. By contrast, the transducer presented in this study handles different vocabularies and manages incompletely aligned datasets.

To address with the diversity of representations of identical entities across different data sources, our proposal utilizes services, such as Balloon [SSB+14] and SameAs (interlinking the Web of Data) [GJM09], that manage co-reference relationships.

Table A.1 presents a comparison of previous studies according to relevant features considered desirable for systems that query the Linked Open Data:

1. Facility for expressing queries. The system offers the users some of the following options when querying heterogeneous datasets of the Linked Open Data: use of keywords, expression of the query using natural language or predefined patterns, or expression of the query using the vocabulary with which the users are familiar.

2. Handling of heterogeneous vocabularies from non-preselected datasets. The system provides answers by accessing datasets that use different vocabularies.

[4] `http://archive.rdf4j.org/users/ch11.html`
[5] `http://lucene.apache.org/solr/`

TABLE A.1: Check for relevant characteristics in related works. (CQ) Conjunctive Queries. (P) Query Patterns. (BGP) Basic Graph Patterns.

	Related work	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**
1	[LFMS12]	√	√	√	√	CQ
2	[LB11]	√	-	-	-	CQ
3	[SMNA14]	√	-	-	-	CQ
4	[GTHP13]	√	√	-	-	P
5	[SS10]	√	-	√	-	CQ
6	[DAB16]	√	-	-	√	CQ
7	[SHH+11b]	√	-	-	-	SPARQL 1.0
8	[QL08b]	√	-	-	-	SPARQL 1.0
9	[FF14]	-	√	-	√	SPARQL 1.1
10	[Har11]	-	-	√	-	SPARQL 1.0
11	[LT10]	-	-	-	-	BGP
12	[TUY11]	-	√	-	-	BGP
13	[MBGC12b]	√	√	√	-	SPARQL 1.0
14	[CSM+10b]	√	√	-	-	BGP
15	[RK13]	-	-	-	√	SPARQL 1.0
16	[DSW06]	-	-	-	√	SeRQL[4]
17	[HMM13]	√	-	-	√	Solr 3.x[5]
18	[FCPW17]	-	-	√	√	SPARQL 1.1
19	Approach in this paper	√	√	√	√	SPARQL 1.0

3. No preprocessing required. The system does not require a costly preprocessing task to create a centralized repository, nor does it build a federation of data sources, a specific index, or require an integration task.

4. Imprecise answers are admitted when a semantics-preserving transformation cannot be guaranteed. Incomplete mappings are supported by the system, and transformations that do not preserve semantics are also considered.

5. Query expressivity.

The approach presented in this study satisfies the first four features and supports SPARQL 1.0 expressivity. Blank nodes are not considered; however, this is not a relevant restriction, because in practice, each blank node in a SPARQL query can be replaced by a new variable. [LFMS12] also covers the first four features; however, the expressiveness of the queries considered is quite low based on their generation from keywords provided by the users. Moreover, that method suffers from scalability issues and high processing times. By contrast, none of these are issues in the approach presented here. Previous studies (entries 1–6 in Table A.1) focus on admitting easy informal inputs as queries (e.g., keywords), which reduces their formal counterpart-query expressivity to conjunctive queries (CQs; except for study number 4, which utilized prewritten patterns). [MBGC12b] also covers the first three features and supports SPARQL 1.0 expressivity; however, its restriction to semantics-preserving rewriting greatly narrows its applicability, as can be seen from the evaluation presented in Section A.5.

Other previous studies (entries 7–12 in Table A.1) utilize a distributed query processing approach and support notable query expressivity: SPARQL 1.1 (study number 9), SPARQL 1.0 (studies 7, 8, and 10), and basic graph

patterns (BGPs) [6] (study numbers 11 and 12). All of these studies (except study number 10) require a costly preprocessing task to build a federation or a specific index. Moreover, all of them (except study numbers 9 and 12) address datasets that share the query vocabulary. Other previous work (study numbers 15–18) consider non-semantics-preserving query transformations and support notable query expressivity; however, they only consider a single vocabulary for the query.

A.3 Preliminaries

In this section, we briefly introduce key concepts and notations used throughout the remainder of the paper. For a complete definition of RDF and SPARQL, we refer to [CWL14, SWG13a].

Let I be the set of all *IRIs*, L be the set of RDF *literals*, and B be the set of RDF *blank nodes*. These three infinite sets are pairwise disjoint. An RDF *triple* is a tuple $(s, p, o) \in (I \cup B) \times I \times (I \cup B \cup L)$. s is called the *subject*, p is called the *predicate*, and o is called the *object* of the triple. A finite set of triples can be represented as a directed-edge-labelled graph, where subjects and objects are nodes, and edges are labelled by predicates. An RDF *graph* is a finite set of triples. For the purpose of this study, a *dataset* is an RDF graph. Given a dataset D, we refer to the set $voc(D) \subseteq I$ of IRIs in D as the *vocabulary* of D.

SPARQL is the standard query language for RDF. The core of a SPARQL query is a *graph pattern*, which is used to match an RDF graph in order to search for the required answers. Let V be an infinite set of *variables* disjoint from $(I \cup B \cup L)$. Variables in V are denoted by prefixing them with a question mark (i.e., $?x$). In this study, a *triple pattern* is a tuple in $(I \cup V) \times (I \cup V) \times (I \cup L \cup V)$. This suggests that a triple pattern is a triple without blank nodes, where a variable might occur anywhere within the triple. A set of triple patterns is a *BGP*, and complex graph patterns are formed by using SPARQL operators that combine BGPs (see [SWG13b]). This algebraic notation is used to present the rewriting rules.

Here, a query is represented by its graph pattern. Given a graph pattern P, $voc(P) \subseteq I$ denotes the set of IRIs occurring in P.

Definición 5 *An IRI, t, is* adequate *for a dataset, D, if $t \in voc(D)$.*

A graph pattern, P, is adequate *for a dataset, D, if $voc(P) \subseteq voc(D)$.*

A query is adequate *for a dataset, D, if its graph pattern is adequate for D.*

Our goal was to present and evaluate a process that takes a given query, P_s, adequate for a *source* dataset, D_s, and transforms it into another query, P_t, adequate for a target dataset, D_t. We only addressed the primary problem of a single target dataset, although the process can be iterated using a different target dataset each time. An eventual decomposition of a source query into subqueries and their distribution to different target datasets will be the goal of future work. However, it should be

[6] A set of triple patterns along with optional filters.

noted that such a problem could be solved by a combination of solutions to the primary problem. The transformation process produces P_t as a semantically equivalent query to P_s as long as enough equivalence mappings between $voc(D_s)$ and $voc(D_t)$ are found. These mappings can originate from any accessible device, including VoID[7] linksets, co-reference services, and mapping services. The distinguishing factor is that the process produces a mimetic query, P_t, even in the case when no equivalent translation for P_s is found. This indicates that sometimes the transformation does not preserve semantics due to the goal of the transducer being the production of a query adequate for the target dataset demanded by the user. The process is based on graph-pattern-rewriting rules that will be presented in the next section.

A.4 Rule-based transducer

This section presents the set of rules devised to rewrite a graph pattern in a stepwise manner. The goal of the rewriting process is to obtain a new expression of the query that can be properly evaluated within a targeted dataset and different from the source dataset of the original query.

The rule system was devised from a pragmatic viewpoint. The rules establish common-sense heuristics to obtain transformations, even when no semantically equivalent translations are available (e.g., due to vocabulary mismatch). The rules can be easily implemented and efficiently processed, as described in Section A.5. Moreover, preconditions for the application of the rules accounted for a carefully restricted context of the IRIs occurring in the graph pattern. Although restricted, the system was quite effective at achieving transformations (see Section A.5). Nevertheless, the system can be easily extended with additional rules.

We considered five types of rules, with each based on a different motif: equivalence, hierarchy, answer-based, profile-based, and feature-based. We also considered a pragmatic scenario in which a bridge dataset could be considered. To favor the possibility of finding alignments, we admitted mappings between both the source (D_s) and target (D_t) datasets and a bridge (D_b) dataset. That scenario is quite frequent, given that almost any domain contains a popular dataset that might play a role as a reference (e.g., OLiA in the linguistic domain, SNOMED in the clinical domain, reference.data.gov.uk in the governmental domain, or DBpedia in cross-domains).

Mapping repositories and online alignment services were central tools for the development of the proposed transducer. Application of the rules was heavily dependent upon the disposability of alignments. Mapping repositories can be found on the web (e.g., in the DBpedia project[8], Bioportal[9], or data dumps of Freebase/Wikidata mappings[10]). Moreover,

[7] http://www.w3.org/TR/void/
[8] http://wiki.dbpedia.org/services-resources/interlinking
[9] https://bioportal.bioontology.org/mappings
[10] https://developers.google.com/freebase/

some datasets contain associated VoID linksets or incorporate sets of alignments with other domain-sharing datasets (e.g., Jamendo[11] with Geonames and MusicBrainz). There are systems, such as Laundromat[12], that offer application program interfaces for searching and downloading curated mapping datasets. The web service http://sameas.org/ offers equivalent IRIs from different datasets and also co-references stores from third-party projects[13]. Some other systems similar to Balloon[14] or Scarlet [SdM08] can be also useful. Implementations of this transducer should incorporate appropriate access to such sources of knowledge, although this process should be transparent to the user.

The rewriting rules are presented in a sequential style. We named the *context* C to the union of the three graphs D_s, D_t, and D_b plus the graphs of their respective term alignments pair sets (D_s, D_t), (D_s, D_b), and (D_b, D_t). Such a union does not need to be materialized in a single graph. It is enough to be currently accessible by the system in a distributed manner.

We used Q to denote the query being rewritten, QP to refer to a triple pattern in Q, and RP to denote a surrogate graph pattern that will replace the triple pattern QP. Finally, we used CG to denote a *Context Graph* that relates terms occurring in QP and those from the surroundings of these terms occurring in the context C.

The rewriting rule

$$\frac{CG}{Q[QP] \Longrightarrow Q[RP]}$$

indicates that when the context graph, CG, matches the context C (i.e., the triples in CG are present in the context C), the query $Q[QP]$ (that includes the pattern QP) is rewritten into the query $Q[RP]$ by replacing all occurrences of the pattern QP in Q with the pattern RP. Triple patterns are expressed with the notation (s, p, o), which is described in Section A.3, using subscripts when necessary. The letter u in a triple pattern denotes an inadequate IRI (unknown) for the target dataset.

The goal of CG is to determine terms that will be used to form the surrogate graph RP. To identify the triples expressed in CG, the system runs SPARQL queries over C that are directly created from the CG expression. This requires constructing a query, where symbols for terms in the target dataset (i.e., $t{:}\ u_i$) (respectively, bridge dataset, $b{:}\ u_i$) are replaced by query variables to be bound in the target (respectively, bridge) dataset, and symbols used for joins are also replaced by variables producing joins. Therefore, an abstract context graph *(u, p, v) (v, q, t:u)* would produce the query:

```
SELECT  ?tu
WHERE {
 u p ?v .
 ?v q ?tu .
}
```

[11] http://dbtune.org/jamendo/
[12] http://lodlaundromat.org/
[13] http://sameas.org/store/
[14] http://schlegel.github.io/balloon/balloon-fusion.html

Notice that the trigger of the rule is the appearance of the pattern *QP* in the query *Q*, which includes an IRI inadequate for the target dataset. The SPARQL query corresponding to the context graph *CG* is then evaluated over the context *C*, and their bindings are used to form the *RP* that will replace *QP*. It is important to note that the queries corresponding to the context graph *CG* are evaluated over SPARQL endpoints provided by the organizations maintaining the datasets, which come with their own entailment regime for query evaluation. The rules presented in this study only assume the simple entailment regime; however, if a SPARQL endpoint employs a more powerful entailment regime, the rules take advantage of that.

RDF literals present in the queries are not rewritten along with the rules, but with properly constructed string-transformation functions. Previous to the rules-application process, every literal, l, occurring in the graph pattern of the original query is replaced by the corresponding literal $f_t(l)$ in the target dataset, D_t. These transformation functions are built into the system and are associated with each dataset according to its policies for representation of literals (i.e., whether they use typed literals or language-tagged literals). Implementation of transformation functions is based on the proper SPARQL functions STRDT and STRLANG. Therefore, a transformation function for English DBpedia applied to literal `"Gravity"` results in $f_{DBpedia}$(`"Gravity"`) = STRLANG(`"Gravity"`, "en") = `"Gravity"@en`.

We are aware that using *QP* as a graph pattern instead of a triple pattern increases the capabilities of the system. Therefore, the current transducer utilizes this possibility when EDOAL rules (in the category of equivalence rules) apply to graph patterns and not only to triple patterns. Nevertheless, we considered that the results obtained in reference to triple patterns were promising, and a full treatment of such an extended scope will require careful consideration in future work.

Equivalence rules will be the first ones applied and achieve a full semantics-preserving translation of the source query in cases involving enough equivalent mappings of the inadequate IRIs of the query. However, if this is not the case, the distinguishing feature of the proposed rule system is that it replaces the remaining inadequate triple patterns with adequate graph patterns that approximate the original patterns and, therefore, outperforms these other systems that abandon the task without providing answers or those that suppress the inadequate portions of the query. Hierarchy rules might relax or restrict the constraints of the query depending on the mappings used in the rewriting; therefore, the target query might decrease in correctness or completeness (relative to the source query). The latter issues are more difficult to qualify when answer-based, profile-based, and feature-based rules are applied. In fact, both correctness and completeness might suffer from their applications, and it is impossible to know with certainty, because this depends upon the circumstances of the context. The resulting rewritten product can only be traced and analyzed after the actual application of a rule; however, this would be too much of a burden for the layperson user. The ability to provide a quality measurement of the

target query would be very interesting in order to inform the user about the expectations of the resulting answers.

The rewriting algorithm applies the rules on a kind-by-kind basis. Within each kind of rules, the algorithm repeats the application of each rule until no more application is possible. The rules of a kind are sequentially numbered and applied in that numbered sequence. Rules are applied as shown in Algorithm 1. After application of a feature-based kind rules (because new terms can be added to the query), if any inadequate IRI remains in the query, equivalent and hierarchy rules are attempted one more time, after which any triple pattern presenting an inadequate IRI is removed from the query. As soon as a rewriting step returns a query that is adequate for the target dataset, the algorithm stops and returns the produced query.

ALGORITHM 1: Rewriting algorithm

Input: C, Q_s
Output: Q_t
$Q_t \leftarrow Q_s$
if $voc(Q_t) \nsubseteq voc(D_t)$ **then**
 | $Q_t \leftarrow$ APPLY(EquivalenceRules, Q_t, C)
end
if $voc(Q_t) \nsubseteq voc(D_t)$ **then**
 | $Q_t \leftarrow$ APPLY(HierarchyRules, Q_t, C)
end
if $voc(Q_t) \nsubseteq voc(D_t)$ **then**
 | $Q_t \leftarrow$ APPLY(AnswerBasedRules, Q_t, C)
end
if $voc(Q_t) \nsubseteq voc(D_t)$ **then**
 | $Q_t \leftarrow$ APPLY(ProfileBasedRules, Q_t, C)
end
if $voc(Q_t) \nsubseteq voc(D_t)$ **then**
 | $Q_t \leftarrow$ APPLY(FeatureBasedRules, Q_t, C)
end
if $voc(Q_t) \nsubseteq voc(D_t)$ **then**
 | $Q_t \leftarrow$ APPLY(EquivalenceRules, Q_t, C)
end
if $voc(Q_t) \nsubseteq voc(D_t)$ **then**
 | $Q_t \leftarrow$ APPLY(HierarchyRules, Q_t, C)
end
$Q_t \leftarrow$ deleteNonAdequateTriplePatterns($Q_t, voc(D_t)$)
return Q_t

In the following subsections, each kind of rule is explained and motivated using an example. Generic *v:*-prefixed letters *v:s, v:p, v:o* are used to refer to terms belonging to a vocabulary $voc(D_v)$. Specifically, *s:*-prefix, *t:*-prefix, and *b:*-prefix refer to the source, target, and bridge vocabulary, respectively. No relationship is assumed among *s:u, t:u* and *b:u* (although they share the symbol *u*).

To illustrate the behavior of the rules, we considered the following scenario: imagine a film reporter wants to know the names of people who

ALGORITHM 2: APPLY algorithm

Input: *RuleSet*, Q, C
Output: Q
for *each* $r \in RuleSet$ **do**
  while $applicable(r, Q, C)$ **do**
    $Q \leftarrow rewriteWith(r, Q, C)$
  end
end
return Q

acted in a film entitled `Gravity`. Because the reporter is familiar with the vocabulary of LinkedMDB[15] (a linked open dataset about movies), the following SPARQL query is written and submitted to the system and specifying LinkedMDB as the source dataset.

```
PREFIX dc:<http://purl.org/dc/terms/>
PREFIX movie:<http://data.linkedmdb.org/resource/movie/>
SELECT ?actor ?name WHERE
{  ?film dc:title  'Gravity' .
   ?film movie:actor ?actor .
   ?actor movie:actor_name ?name. }
```

The system issues the query to the corresponding SPARQL endpoint; however, unfortunately, no element is received as an answer[16]. The reporter then issues that query to the DBpedia dataset. During a parsing of the query, the IRIs `dc:title`, `movie:actor`, and `movie:actor_name` are discovered as inadequate for DBpedia. The rule-application process is then launched.

A.4.1 Equivalence rules: E1, E2, E3, E4, E5, E6, and E7

Equivalence rules are applied when inadequate IRIs occurring in the query are involved in equivalence mappings (using predicates, such as `owl:sameAs`, `owl:equivalentClass`, `owl:equivalentProperty`) or structural-equivalence mappings, such as these captured by the EDOAL[17] language. The aim of this kind of rule is to transform a query into an equivalent one.

First, in the presence of alignments captured as EDOAL alignments, where the registered relationship is `Equivalence`, if `ELHS` and `t:ERHS` (i.e., every IRI in `ERHS` is in D_t) are, respectively, the left- and right-hand side of an EDOAL equivalence rule, and `ELHS` matches a portion of the graph pattern of the query, then the system replaces the portion matched by `ELHS` in the query with `t:ERHS` (Rule $E1$).

Second, individual equivalence mappings between adequate and inadequate terms are accounted for by rules $E2$, $E3$, and $E4$. The generic predicate *eq* represents a wildcard for any predicate of an extensible set of

[15] http://linkedmdb.org/
[16] The query was issued on 2015-04-15.
[17] http://alignapi.gforge.inria.fr/edoal.html

$$(E1)\quad \frac{EDOAL : ELHS \rightarrow t{:}ERHS}{Q[ELHS] \Longrightarrow Q[ERHS]}$$

$$(E2)\quad \frac{(u, eq, t{:}u_i)(i = 1..n)}{Q[(u, p, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}(t{:}u_i, p, o)]}$$

$$(E3)\quad \frac{(u, eq, t{:}u_i)(i = 1..n)}{Q[(s, u, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}(s, t{:}u_i, o)]}$$

$$(E4)\quad \frac{(u, eq, t{:}u_i)(i = 1..n)}{Q[(s, p, u)] \Longrightarrow Q[\text{UNION}_{i=1..n}(s, p, t{:}u_i)]}$$

$$(E5)\quad \frac{(u, eq, b{:}u_i)(i = 1..n)(b{:}u_i, eq, t{:}u_{ij})(j = 1..n_i)}{Q[(u, p, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}^{j=1..n_i}(t{:}u_{ij}, p, o)]}$$

$$(E6)\quad \frac{(u, eq, b{:}u_i)(i = 1..n)(b{:}u_i, eq, t{:}u_{ij})(j = 1..n_i)}{Q[(s, u, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}^{j=1..n_i}(s, t{:}u_{ij}, o)]}$$

$$(E7)\quad \frac{(u, eq, b{:}u_i)(i = 1..n)(b{:}u_i, eq, t{:}u_{ij})(j = 1..n_i)}{Q[(s, p, u)] \Longrightarrow Q[\text{UNION}_{i=1..n}^{j=1..n_i}(s, p, t{:}u_{ij})]}$$

FIGURE A.1: Equivalence Rules.

equivalence predicates, such as {`owl:sameAs`, `owl:equivalentProperty`, `owl:equivalentClass`, and `skos:exactMatch`}. Our implementation addresses *eq* as a symmetric property; therefore, triple *(x,eq,y)* means the same as triple *(y,eq,x)*, but it does not assume the transitive closure of *eq* to be computed in the dataset graph, and does not assume enough of a SPARQL entailment regime to infer such transitivity.

For the purposes of an example implementation, the search for triples of the form *(u,eq,t:u)* in the context C, where u is `dc:title`, `movie:actor`, or `movie:actor_name`, is implemented by parameterized SPARQL queries over the corresponding mapping repositories. The service sameAs.org reported 12 synonyms for `dc:title`, with two of them (`foaf:name` and `rdfs:label`) adequate for DBpedia. However, the service did not report any adequate synonyms for `movie:actor` or `movie:actor_name`. Applying rule E3, the graph pattern was transformed into the following:

```
{{?film rdfs:label 'Gravity'@en.} UNION {?film foaf:name 'Gravity'@en.}
 ?film movie:actor ?actor .
 ?actor movie:actor_name ?name. }
```

Third, a selected bridge dataset, D_b, can be considered helpful for identifying additional equivalence mappings. This is enabled by rules $E5$, $E6$, and $E7$. Note that the context graph is fulfilled with the evaluation of a SPARQL query over the corresponding SPARQL endpoint. In the case of the example, the search for triples of the form (`movie:actor`,*eq,b:u*) and

(*b:u,eq,t:u*) using Freebase[18] as bridge dataset did not succeed. Therefore, the application of equivalence rules finished, and the next step involved the application of hierarchy rules.

A.4.2 Hierarchy rules: H8, H9, H10, H11, H12, and H13

These rules transform the query by generalizing or specializing inadequate IRIs for which equivalence rules failed to translate. The aim of these rules is to construct a looser or tighter query when the known mappings do not inform direct equivalences.

We used a generic predicate *sub* to represent a wildcard for any predicate of an extendible set of hierarchy predicates, such as {`skos:narrower`, `skos:broader`, `rdfs:subClassOf`, and `rdfs: subPropertyOf`}. The system considers two basic possibilities: 1) when an IRI in a triple pattern is known to be in a *"subterm"* relationship with a collection of adequate IRIs, the triple pattern will be replaced by the conjunction (i.e., AND operator) of looser triple patterns (rules *H*8–*H*10); and 2) when an IRI in a triple pattern is known to be in a *"superterm"* relationship with a collection of adequate IRIs, the triple pattern will be replaced by the disjunction (i.e., UNION operator) of tighter triple patterns (rules *H*11–*H*13). The difference between rules *Hxa* and *Hxb* (see Figure A.2) is that the context graph in *Hxa* rules only considers triples relating the unknown u and a target term t: u_i directly with a *sub* property; however, the context graph in *Hxb* rules considers composition of two triples with the *sub* property to relate the unknown u to a target term t: u_i.

The set of hierarchy rules is very simple, and it can be argued that additional possibilities for the context-graph precondition could be considered. This could involve EDOAL hierarchy rules or combinations of predicates *sub*, with predicate *eq* or concatenation of additional *sub* predicates. However, we settled for these, because they provided a good balance between simplicity and added benefits. We do not claim that the present selection of rules is the best. In fact, we were more interested in demonstrating the simplicity and extensibility of the proposed approach to SPARQL query rewriting and considered the implementation presented here as a baseline for future improvements.

With respect to the example, the system found the mapping (`movie: actor_name`, `rdfs:sub PropertyOf`, and `foaf:name`) in a mapping repository. The application of rule H9a transformed the query into the following:

```
{{?film rdfs:label 'Gravity'@en.} UNION {?film foaf:name 'Gravity'@en.}
 ?film movie:actor ?actor.
 ?actor foaf:name ?name. }
```

The IRI `movie:actor` remains inadequate for DBpedia.

[18] https://www.freebase.com/

$$(H8a)\quad \frac{(u, sub, t{:}u_i)(i = 1..n)}{Q[(u, p, o)] \Longrightarrow Q[\text{AND}_{i=1..n}(t{:}u_i, p, o)]}$$

$$(H8b)\quad \frac{(u, sub, v_i)(i = 1..n)(v_i, sub, t{:}u_{ij})(j = 1..n_i)}{Q[(u, p, o)] \Longrightarrow Q[\text{AND}_{i=1..n}^{j=1..n_i}(t{:}u_{ij}, p, o)]}$$

$$(H9a)\quad \frac{(u, sub, t{:}u_i)(i = 1..n)}{Q[(s, u, o)] \Longrightarrow Q[\text{AND}_{i=1..n}(s, t{:}u_i, o)]}$$

$$(H9b)\quad \frac{(u, sub, v_i)(i = 1..n)(v_i, sub, t{:}u_{ij})(j = 1..n_i)}{Q[(s, u, o)] \Longrightarrow Q[\text{AND}_{i=1..n}^{j=1..n_i}(s, t{:}u_{ij}, o)]}$$

$$(H10a)\quad \frac{(u, sub, t{:}u_i)(i = 1..n)}{Q[(s, p, u)] \Longrightarrow Q[\text{AND}_{i=1..n}(s, p, t{:}u_i)]}$$

$$(H10b)\quad \frac{(u, sub, v_i)(i = 1..n)(v_i, sub, t{:}u_{ij})(j = 1..n_i)}{Q[(s, p, u)] \Longrightarrow Q[\text{AND}_{i=1..n}^{j=1..n_i}(s, p, t{:}u_{ij})]}$$

$$(H11a)\quad \frac{(t{:}u_i, sub, u)(i = 1..n)}{Q[(u, p, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}(t{:}u_i, p, o)]}$$

$$(H11b)\quad \frac{(v_i, sub, u)(i = 1..n)(t{:}v_{ij}, sub, v_i)(j = 1..n_i)}{Q[(u, p, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}^{j=1..n_i}(t{:}v_{ij}, p, o)]}$$

$$(H12a)\quad \frac{(t{:}u_i, sub, u)(i = 1..n)}{Q[(s, u, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}(s, t{:}u_i, o)]}$$

$$(H12b)\quad \frac{(v_i, sub, u)(i = 1..n)(t{:}v_{ij}, sub, v_i)(j = 1..n_i)}{Q[(s, u, o)] \Longrightarrow Q[\text{UNION}_{i=1..n}^{j=1..n_i}(s, t{:}v_{ij}, o)]}$$

$$(H13a)\quad \frac{(t{:}u_i, sub, u)(i = 1..n)}{Q[(s, p, u)] \Longrightarrow Q[\text{UNION}_{i=1..n}(s, p, t{:}u_i)]}$$

$$(H13b)\quad \frac{(v_i, sub, u)(i = 1..n)(t{:}v_{ij}, sub, v_i)(j = 1..n_i)}{Q[(s, p, u)] \Longrightarrow Q[\text{UNION}_{i=1..n}^{j=1..n_i}(s, p, t{:}v_{ij})]}$$

FIGURE A.2: Hierarchy Rules.

A.4.3 Answer-based rules: A14, A15, A16, A17, A18, and A19

These rules use bindings of variables (not necessarily those in the SELECT clause) obtained during query evaluation over the source dataset as examples of what the source query is looking for in the target dataset. Triples in the target dataset involving equivalent terms to those bound to variables are used to mimic the triple pattern to be replaced. The reason is that triples in the target dataset and concerning the terms the variables were bound to likely resemble the expected bindings of the original query over that target dataset.

Note that only a restricted set of triple patterns is selected for application of the answer-based rewriting rules due to the uncertain scenario they address. Of course, at least one variable must appear in the pattern to ensure meaningful application of the answer-based rules.

If only one variable appears in the triple pattern (see rules *A*14–*A*17), an adequate IRI for the target dataset is required for any of the two other triple positions. The reason for this requirement is that in this context, an adequate IRI represents an anchor point for reducing uncontrolled rewritings. The adequate IRI might exist from the beginning (i.e., some IRIs can be adequate for both source and target datasets) or the result of a previously applied rewriting rule. In any case, we denote the IRI that appeared in the source query before any rewriting as $R^{-1}(t\colon p)$. Therefore, *Answers(?x, t:p, u)* (in *A*14), where u is an adequate IRI for the source dataset, but inadequate for the target dataset, is the binding set resulting from evaluation of the following query over the source dataset:

`SELECT ?x WHERE {?x` R^{-1}`(t:p) u.}`

Analogously, this also holds for any of the other triple patterns (*(u, t:p, ?x)*, *(?x, u, t:o)*, and *(t:s, u, ?x)*) appearing in the *QP* of answer-based rules *A*15 through *A*17.

In general, rules *A*14 and *A*15 apply as follows: triple patterns with a variable in the subject (respectively object) with adequate predicate *t:p* and an inadequate object (respectively subject) will be replaced by the union of conjunctions of triple patterns comprising samples of objects (respectively subjects) related to answers (after *eq* predicate) according to the predicate *t:p*.

Note that such a precondition involves a nested SPARQL query, where the bindings obtained with the *Answers( )* query over the source dataset (denoted by $[x_1, \ldots, x_n]$) are used in the precondition context graph.

Rules *A*16 and *A*17 apply as follows: triple patterns with a variable in the subject (respectively object) with an inadequate predicate, u, and an adequate object *t:o* (respectively subject) will be replaced by the conjunction of triple patterns comprising the shared predicates of the triples, where answers (after the *eq* predicate) are subjects and *t:o* is the common object of all these triples. This implies that the triple pattern with the inadequate predicate, u, is replaced by the conjunction of triple patterns formed with the r adequate predicates $(t\colon p_k)_{k=1..r}$, for which the terms from the bindings $(t\colon x_{ij})_{i=1..n}^{j=1..n_i}$ (after *eq* predicate) agree on their object

value *t:o* (respectively subject value *t:s*). Note that the significant replacement of a predicate with so little information is audacious. Previous studies addressed instance matching [CFMV11]; however, interlinking similar properties occurs infrequently and remains a real challenge [ZGB+17]. In fact, scarcity of property alignments is a reason why patterns, QP, with only one variable in the predicate position are not considered in answer-based rules. However, a more relevant reason to avoid such patterns is that the application of analogous replacements to those in rules $A14$ through $A17$ would generally produce graph patterns hardly similar to the original.

Additional forms of predicate replacement are proposed in subsequent rule classes. We are aware that more complex techniques can be used to deal with this issue; however, it must be also considered that execution time and knowledge-resource availability are hard constraints of the scenario addressed by our transducer.

When two variables appear in the triple pattern (see rules $A18$ and $A19$), the remaining term is the inadequate IRI required in the subject or object position due to the pragmatic consideration that bindings of object or subject variables are more likely to serve as examples for replacing what is unknown in the triple pattern than bindings of predicate variables, which carry more semantics and would be riskier to replace.

Therefore, rules $A18$ and $A19$ are applied to patterns of the style $(?x, ?p, u)$ and $(u, ?p, ?x)$, respectively. The replacement will be another triple pattern of the same style, with the inadequate object (respectively subject) substituted with an adequate IRI, which is the most frequent object (respectively subject) in triples from the target dataset describing the resources of the bindings.

More specifically, *mostFrequent(?x, ?p, u)* denotes the result of the following nested query [and analogously for *mostFrequent(u, ?p, ?x)*]:

```
PREFIX s:<source dataset>
SELECT ?o (COUNT(?o) as ?OCCURNUM)
FROM <target dataset>
WHERE {
   ?xt  ?pt  ?o .
   {
   SELECT ?xt
   FROM <mappings dataset>
   WHERE {
      ?xs  eq  ?xt .
      {
      SELECT ?xs
      FROM  <source dataset>
      WHERE { ?xs  ?ps  s:o . }
      }
   }
   }
}
GROUP BY ?o ORDER BY DESC ?OCCURNUM
LIMIT 1
```

$$(A14)\quad \frac{Answers(?x, t{:}p, u) = [x_1, \ldots, x_n] \quad (i = 1..n)(x_i, eq, t{:}x_{ij})(j = 1..n_i)(t{:}x_{ij}, t{:}p, t{:}o_{ij}^{k})(k = 1..n_{ij})}{Q[(?x, t{:}p, u)] \Longrightarrow Q[\text{UNION}_{i=1..n}(\text{AND}_{j=1..n_i}^{k=1..n_{ij}}(?x, t{:}p, t{:}o_{ij}^{k}))]}$$

$$(A15)\quad \frac{Answers(u, t{:}p, ?x) = [x_1, \ldots, x_n] \quad (i = 1..n)(x_i, eq, t{:}x_{ij})(j = 1..n_i)(t{:}s_{ij}^{k}, t{:}p, t{:}x_{ij})(k = 1..n_{ij})}{Q[(u, t{:}p, ?x)] \Longrightarrow Q[\text{UNION}_{i=1..n}(\text{AND}_{j=1..n_i}^{k=1..n_{ij}}(t{:}s_{ij}^{k}, t{:}p, ?x))]}$$

$$(A16)\quad \frac{Anwers(?x, u, t{:}o) = [x_1, \ldots, x_n] \quad (i = 1..n)(x_i, eq, t{:}x_{ij})(j = 1..n_i)(t{:}x_{ij}, t{:}p_k, t{:}o)(k = 1..r)}{Q[(?x, u, t{:}o)] \Longrightarrow Q[\text{AND}_{k=1..r}(?x, t{:}p_k, t{:}o)]}$$

$$(A17)\quad \frac{Anwers(t{:}s, u, ?x) = [x_1, \ldots, x_n] \quad (i = 1..n)(x_i, eq, t{:}x_{ij})(j = 1..n_i)(t{:}s, t{:}p_k, t{:}x_{ij})(k = 1..r)}{Q[(t{:}s, u, ?x)] \Longrightarrow Q[\text{AND}_{k=1..r}(t{:}s, t{:}p_k, ?x)]}$$

$$(A18)\quad \frac{mostFrequent(?x, ?p, u) = [t{:}o]}{Q[(?x, ?p, u)] \Longrightarrow Q[(?x, ?p, t{:}o)]}$$

$$(A19)\quad \frac{mostFrequent((u, ?p, ?x)) = [t{:}s]}{Q[(u, ?p, ?x)] \Longrightarrow Q[(t{:}s, ?p, ?x)]}$$

FIGURE A.3: Answer-Based Rules.

Resuming the query example, answer-based rules A14 through A19 do not apply, because the pattern portion, QP, of all of these rules [(*?x, t:p, u*), (*u, t:p, ?x*), (*?x, u, t:o*), (*t:s, u, ?x*), (*?x, ?p, u*), and (*u, ?p, ?x*)] does not match the triple pattern `(?film, movie:actor, ?actor)`. However, to present an example of the application of these types of rules, consider another query where (`movie:film/1116, dbo:producer, ?x`) is part of the graph pattern with `movie:film/1116` as the only inadequate IRI, and the only binding for `?x` in the source dataset is `movie:producer/9659`. Moreover, let us assume the following triples:
(`movie:producer/9659, owl:sameAs, dbr:Charlie_Chaplin`),
(`dbr:The_Gold_Rush, dbo:producer, dbr:Charlie_Chaplin`),
and (`dbr:The_Great_Dictator, dbo:producer, dbr:Charlie_Chaplin`) are present in the context C. Therefore, rule A15 would be used to transform the pattern (`movie:film /1116, dbo:producer, ?x`) into the following: {`dbr:The_Gold_Rush, dbo:producer, ?x`} `AND` {`dbr:The_Great_Dictator, dbo:producer, ?x`}. The transducer substitutes the inadequate IRI `movie:film/1116` with samples of adequate IRIs appearing in the target dataset and attributed to the same IRI (`dbr:Charlie_Chaplin`) for the corresponding property (`dbo:producer`).

A.4.4 Profile-based rules: P20, P21, and P22

These rules consider the triples in a dataset describing the resource denoted by an IRI in the graph pattern. In this study, the *profile* of an IRI, x, in a dataset, D, comprises three sets of IRIs related to x by triples in D. The set $\mathcal{F}_D(x)$ gathers the facets of x (the IRIs of predicates of triples in D, where x participates as subject or object). The set $\mathcal{V}_D(x)$ gathers the IRIs that are values of the facets of x, and the set $\mathcal{E}_D(x)$ gathers the subjects or objects of triples in D, where x is the predicate.

$$\mathcal{F}_D(x) = \{p \in voc(D) \quad | (\exists v.(x,p,v) \in D \vee (v,p,x) \in D)\}$$

$$\mathcal{V}_D(x) = \{v \in voc(D) \quad | (\exists p.(x,p,v) \in D \vee (v,p,x) \in D)\}$$

$$\mathcal{E}_D(x) = \{v \in voc(D) \quad | (\exists w.(v,x,w) \in D \vee (w,x,v) \in D)\}$$

The situation addressed using these rules is as follows: if an IRI, v, in the (values part of the) profile of an IRI, u, inadequate for the target dataset denotes the same resource as that denoted by the IRI *t:v* in the target dataset, and if there is an IRI *t:u* in the (values part of the) profile of *t:v* denoting a resource sufficiently similar to u, then u will be replaced by *t:u*.

A similarity function $[\phi(u,w)]$ between resources denoted by IRIs was defined, and we established a threshold, h, to discard all of these IRIs, w, with similarity values not surpassing the threshold and considering them insufficiently similar to u. From the IRIs that passed the filter, that with the maximum similarity value was chosen.

Usually the comparison of resources denoted by two different IRIs is based on the proper context of each IRI. In this case, the context of an IRI is the set of triples where it appears. These triples inform different facets of the denoted resource: some of them 1) associate text descriptions to the resource (`rdfs:label`, `rdfs:comment`) or other annotation properties, 2) describe relationships to other resources, or 3) describe hierarchical or membership relationships. Therefore, different opportunities exist to define a similarity based on that context. We decided to select three facets and combine them linearly to balance the contribution of each one. This idea has been successfully used in previous studies [TGEM07], and our proposal includes a contrasted state-of-the-art metric for each facet. Although it is possible to experiment with alternatives, the results obtained using our selection are promising (see Section A.5).

Let us denote using $maxSim(u, h, R)$ an IRI, w, in the set of IRIs, I, which is the most similar value to u and whose similarity value is greater

than the threshold value h:

$$maxSim(u, h, R) = w \quad \text{such that} \quad \begin{array}{l} w \in I, \\ \phi(u, w) \geq h, \quad \text{and} \\ \forall v \in I.\ \phi(u, w) \geq \phi(u, v) \end{array}$$

where the similarity of a pair of IRIs, $\phi(u, v)$, is calculated as a linear combination of three basic similarity measures, (S_n, S_d, and S_o) selected from the state-of-the-art in order to cover various features (name, description, or ontological context) attributed to an IRI.

Two of them (S_n and S_d) are string-based methods supported by string-valued properties representing a name and a description, respectively, of the resources denoted by the IRIs. Another one, S_o, considers the ontological context of the IRIs.

$$\begin{aligned} \phi(u, v) \quad &= \alpha_n \cdot S_n(u, v) + \alpha_d \cdot S_d(u, v) + \alpha_o \cdot S_o(u, v) \\ &\alpha_n, \alpha_d, \alpha_o \geq 0 \quad \wedge \quad \alpha_n + \alpha_d + \alpha_o = 1 \end{aligned}$$

S_n is a language-based similarity measure computed using a formula from Wu and Palmer [WP94] that calculates the relatedness of the two Wordnet synsets corresponding to the `rdfs:label` property value of the two compared IRIs. The system relies on natural-language-processing techniques to perform tasks, such as linguistic normalization, lemmatization, term extraction, stop-word elimination, and language detection, over the `rdfs:label` property value, thereby associating a synset to each compared IRI. Multilingual databases are considered (e.g., Extended Multilingual wordnet[19] or WordNet for European languages[20]) to accommodate resources annotated using different languages.

S_d relies upon a common technique in the information-retrieval domain. It is a token-based measurement that accounts for the ontological contexts of the IRIs [Res95]. For each IRI, u and v, a bag of words is constructed containing words from their `rdfs:comment` properties and the `rdfs:label` properties of IRIs related to u and v, respectively, via domain properties. S_d represents the cosine similarity of two vectors, $V(u)$ and $V(v)$, constructed by the frequency of word appearance (i.e., using the vector space model technique).

$$S_d(u, v) = \frac{V(u) \cdot V(v)}{\|V(u)\|\,\|V(v)\|}$$

S_o is a path-distance measurement appearing in [ES13a] and considered only for IRIs representing a class or a property (i.e., if both IRIs represent neither classes nor properties, the value of this measure is zero).

Two paths are considered for each IRI, t: the upward path $p(t) = \langle p_i \rangle_{i=1}^{m}$ and the downward path $d(t) = \langle d_i \rangle_{i=1}^{n}$, where $p_m = d_n = t$ and $p_{i-1} = \{r \in$

[19] http://compling.hss.ntu.edu.sg/omw/summx.html
[20] http://projects.illc.uva.nl/EuroWordNet/

TABLE A.2: Statistics for various threshold values used in the experiments: *Num* represents the number of terms, w, such that $\phi(u, w) \geq h$ during the experiments for each h. *Rate* is the percentage that *Num* represents against the 95 total or other terms.

	$h = 0.25$	$h = 0.45$	$h = 0.5$	$h = 0.75$	$h = 0.9$
Num	61	44	41	19	8
Rate	64%	46%	43%	20%	8%

$I|\exists s \in p_i.\ s \subset r\}$ and $d_{i-1} = \{r \in I|\exists s \in d_i.\ s \supset r\}$. We consider the path distance defined as follows:

$$\begin{aligned} \delta(\langle s_i\rangle_{i=1}^{m}, \langle s'_j\rangle_{j=1}^{n}) &= \lambda \times \delta'(s_m, s'_n) + (1-\lambda) \times \delta(\langle s_i\rangle_{i=1}^{m-1}, \langle s'_j\rangle_{j=0}^{n-1}) \\ \delta(\langle\rangle, \langle s'_j\rangle_{j=1}^{k}) &= \delta(\langle s_i\rangle_{i=1}^{k}, \langle\rangle) = k \end{aligned}$$

where $\lambda = 0.5$, and $\delta'(s_m, s'_n)$ is the Levenshtein distance on the string formed by the concatenation of the `rdfs:label` property of the IRIs in sets s_m and s'_n, respectively.

$$S_o(u, v) = \frac{1}{2}\delta(p(u), p(v)) + \frac{1}{2}\delta(d(u), d(v))$$

The selection of concrete values for the threshold, h, and the parameters α_n, α_d, and α_o allows tuning of the similarity level depending on the context. The more strictness used in selecting an IRI denoting a similar resource, the higher h should be. In our experiments, a total of 95 IRIs were considered for the eventual application of a profile rule, resulting in the results shown in Table A.2. We used a threshold value of $h = 0.45$ to balance the quantity of discarded and accepted IRIs.

Value selection for α_n, α_d, and α_o allowed bias in the similarity to favor specific aspects of the resources. Depending on the context, more emphasis could be applied to linguistic aspects or ontological relationships. For the sake of experiments using the selected datasets, we overvalued the ontological aspects and chose the following values: $\alpha_n = 0.1$, $\alpha_d = 0.2$, and $\alpha_o = 0.7$.

According to the example, rules $P20$ and $P21$ did not apply, because the triple pattern QP triggering the rules requires an inadequate IRI subject (for rule $P20$) or an IRI object (for rule $P21$), whereas the triple pattern to be matched `(?film, movie:actor, ?actor)` presents variables in these places. However, it matched the triple pattern according to rule $P22$, and the following triples [21] (among others) appear in LinkedMDB and DBpedia:

```
(movie:film/62333, movie:actor, movie:actor/338)
```

[21] Prefix `db-o:<http://dbpedia.org/ontology/>` `db-r:<http://dbpedia.org/resource/>`

$$(P20)\quad \frac{(i=1..n)(a_i \in \mathcal{V}_{D_s}(u))(a_i, eq, t{:}a_{ij})(j=1..n_i) \quad R = \cup_{i=1..n}^{j=1..n_i} \mathcal{V}_{D_t}(t{:}a_{ij}),\ \ t{:}ms = maxSim(u,h,R)}{Q[(u,p,o)] \Longrightarrow Q[(t{:}ms,p,o)]}$$

$$(P21)\quad \frac{(i=1..n)(a_i \in \mathcal{V}_{D_s}(u))(a_i, eq, t{:}a_{ij})(j=1..n_i) \quad R = \cup_{i=1..n}^{j=1..n_i} \mathcal{V}_{D_t}(t{:}a_{ij}),\ \ t{:}ms = maxSim(u,h,R)}{Q[(s,p,u)] \Longrightarrow Q[(s,p,t{:}ms)]}$$

$$(P22)\quad \frac{(i=1..n)(a_i \in \mathcal{E}_{D_s}(u))(a_i, eq, t{:}a_{ij})(j=1..n_i) \quad R = \cup_{i=1..n}^{j=1..n_i} \mathcal{F}_{D_t}(t{:}a_{ij}),\ \ t{:}ms = maxSim(u,h,R)}{Q[(s,u,o)] \Longrightarrow Q[(s,t{:}ms,o)]}$$

FIGURE A.4: Profile-based Rules.

```
(movie:actor/338, owl:sameAs, db-r:Alastair_Mackenzie)
(db-r:Alastair_Mackenzie, db-o:starring, db-r:The_Last_Great_Wilderness)
(movie:film/1894, movie:actor, movie:actor/40969)
(movie:film/1894, owl:sameAs, db-r:Killer's_Kiss)
(db-r:Killer's_Kiss, db-o:producer , db-r:Stanley_Kubrick)
(movie:film/10849, movie:actor, movie:actor/29437)
(movie:film/10849, owl:sameAs, db-r:The_Indian_Runner)
(db-r:The_Indian_Runner, db-o:director , db-r:Sean_Penn)
```

The similarity function between `movie:actor` and each IRI of the properties set {`db-o:starring, db-o:producer, db-o:director`} were evaluated, resulting in the following outcomes:

$$\begin{aligned}
\phi(\texttt{movie:actor}, \texttt{db-o:starring}) &= 0.1 \cdot 0.71 + 0.2 \cdot 0.89 + 0.7 \cdot 0.97 = 0.934 \\
\phi(\texttt{movie:actor}, \texttt{db-o:producer}) &= 0.1 \cdot 0.58 + 0.2 \cdot 0.82 + 0.7 \cdot 0.96 = 0.896 \\
\phi(\texttt{movie:actor}, \texttt{db-o:director}) &= 0.1 \cdot 0.39 + 0.2 \cdot 0.68 + 0.7 \cdot 0.84 = 0.768
\end{aligned}$$

The greatest similarity was achieved with `db-o:starring`, with this value above the threshold parameter. Therefore, `db-o:starring` substitutes the predicate `movie:actor`, and the graph pattern becomes an adequate query for the DBpedia dataset:

```
{{?film rdfs:label  'Gravity'@en.} UNION {?film foaf:name 'Gravity'@en .}
 ?film db-o:starring ?actor .
 ?actor foaf:name ?name. }
```

The result of its evaluation is shown in Table A.3. The eight first answers were not from the film Gravity, but rather the television series; however, the precision of the result set was adequate, because facets of the searched resource were not explicit in the source query.

Therefore, with respect to our example, feature-based rules were not used, and the rule-application algorithm reached the end. However, the rewriting system accounted for the possibility that inadequate terms could remain in the query.

TABLE A.3: Answers from the adequate query of DBpedia.

?actor	?name
<http://dbpedia.org/resource/Eric_Schaeffer>	Eric_Schaeffer
<http://dbpedia.org/resource/Krysten_Ritter>	Krysten_Ritter
<http://dbpedia.org/resource/Ivan_Sergei>	Ivan_Sergei
<http://dbpedia.org/resource/Ving_Rhames>	Ving_Rhames
<http://dbpedia.org/resource/Rachel_Hunter>	Rachel_Hunter
<http://dbpedia.org/resource/Robyn_Cohen>	Robyn_Cohen
<http://dbpedia.org/resource/James_Martinez_(actor)>	James_Martinez
<http://dbpedia.org/resource/Seth_Numrich>	Seth_Numrich
<http://dbpedia.org/resource/Sandra_Bullock>	Sandra_Bullock
<http://dbpedia.org/resource/George_Clooney>	George_Clooney

$$(F23)\quad \frac{(u, s{:}p_i, s{:}o_i)(i = 1..n)}{Q[(u,p,o)] \Longrightarrow Q[(?v,p,o)\text{AND}_{i=1..n}(?v, s{:}p_i, s{:}o_i)} \quad ?v \text{ a new variable}]$$

$$(F24)\quad \frac{(s{:}s_i, u, s{:}o_i)(i = 1..n)}{Q[(s,u,o)] \Longrightarrow Q[\text{AND}_{i=1..n}(s{:}s_i, ?v, s{:}o_i)} \quad ?v \text{ a new variable}]$$

$$(F25)\quad \frac{(s{:}s_i, s{:}p_i, u)(i = 1..n)}{Q[(s,p,u)] \Longrightarrow Q[(s,p,?v)\text{AND}_{i=1..n}(s{:}s_i, s{:}p_i, ?v)} \quad ?v \text{ a new variable}]$$

FIGURE A.5: Feature-based Rules.

A.4.5 Feature-based rules: F23, F24, and F25

These rules are the last option if inadequate terms remain in the graph pattern after the other rules have already been considered. In this case, the expectation is to replace the inadequate term with a new variable (thereby generalizing the query), but restricting that variable with features of the replaced term (i.e., triples where the replaced term is the subject).

Note that rules $F23$ through $F25$ might introduce new IRIs not previously present in the query, and that some of these could be inadequate for the target dataset. As expressed in Algorithm 1, equivalence and hierarchy rules (from $E1$-$E7$, and from $H8$-$H13$, in that order) are applied to the resulting feature-based transformed query graph. Afterward, any residual inadequate triple pattern is removed from the graph pattern.

A.5 Evaluation

In this section, we evaluate a prototype implementation of the presented transducer to address the following questions:

1. Are transformation rules other than equivalence and hierarchy rules considered relevant?

2. Is the transformation process relevant enough to be maintained in its entirety without skipping any kind of rule?

3. What is the accuracy of the outcome queries produced by the transducer with respect to the expected answers obtained by gold-standard queries?

4. Is the processing time of the implemented prototype acceptable?

First, a query benchmark was required to allow evaluation of transducer behavior. However, we were unable to locate an available benchmark appropriate for testing SPARQL query rewriting of the style proposed in this study. There exist different benchmarks for measuring the query performance against RDF repositories (i.e., benchmarks for testing federated SPARQL query systems [GTS12], for streaming RDF/SPARQL engines [ZDCC12], for generating SPARQL queries out of a set of keywords [SGH+11b], or for testing RDF stores with real queries on real data [MLAN11]); however, there are no benchmarks for challenging a transducer attempting to perform twin queries over shared-domain datasets with different vocabularies. Therefore, we constructed a proper benchmark for that task.

A.5.1 Benchmark generation

The infrastructure considered for the experiments is the Linked Open Data environment leveraging datasets comprising SPARQL endpoints and selecting a set of queries from diverse benchmarks and real logs. Our goal was to consider enough heterogeneity on both the dataset-domain and the query structure sides. We relied upon well-known datasets with available SPARQL endpoints and accessible to people wishing to assess our experiments. The following sections present the selected datasets and the set of queries run on those datasets.

A.5.1.1 Datasets selection

Three domain areas were considered for the datasets: media domain, bibliography, and life science. For each, a set of recognized datasets was selected. With respect to the media domain, we selected: DBpedia, MusicBrainz, LinkedMDB, Jamendo, New York Times, and BBC. With respect to bibliographic domain, we considered BNE (Biblioteca Nacional de España), BNF (Bibliothèque National de France), BNB (British National Bibliography), LIBRIS, DBLP, and Cambridge. For life science, we selected: Drugbank, SIDER, CHEBI, DISEASOME, and KEGG. Moreover, to achieve greater plurality in the tests, we used SP2Bench, which is based on a synthetic dataset.

To perform the queries, we used the available endpoint of each mentioned dataset, except for the SP2Bench queries, we reused an existing available synthetic dataset of 10^6 triples. Therefore, 16 real datasets plus a synthetic one were considered for testing the system, thereby providing evidence of the broad coverage of the tests performed.

TABLE A.4: Number of mappings between pairs of datasets.

Datasets		Number of mappings
DBpedia	LinkedMDB	13800
MusicBrainz	DBpedia	22981
MusicBrainz	Jamendo	15000
DBpedia	New York Times	10359
DBpedia	BBC	76171
DBpedia	Sider	751
DBpedia	Drugbank	729
DBpedia	DBLP	196
DBpedia	Cambridge	1859
DBpedia	BNE	36431
DBpedia	BNB	89452
DBpedia	BNF	41000
DBpedia	SP2B	1250
DBpedia	Libris	10884
Drugbank	KEGG	128
Drugbank	Diseasome	1943
Drugbank	Sider	729
Drugbank	CHEBI	28213
BNE	BNF	9725
BNE	BNB	45000
BNB	VIAF	400000
BNF	VIAF	400000
DBLP	BNB	196

Table A.4 shows the list of pairs of datasets used as sources or targets (interchangeable) in the benchmark along with their corresponding number of mappings between terms in their respective vocabularies and obtained from their corresponding linksets.

A.5.1.2 Query Set selection

When selecting the queries, our aim was to obtain a set containing a broad spectrum of SPARQL query types. For that reason, we looked at two criteria when making our selection: place of provenance of the queries and their syntactic structure.

Tables A.10, A.11, A.12, A.13, A.14, A.15, and A.16 in the Appendix A.7 show the sets of queries grouped by domain and consecutively numbered from Q1 to Q50. For each query, the following information is presented: number, provenance (within brackets), and source dataset. In the box, the text of the source query is presented, and below that, the corresponding gold-standard query for the target dataset (left) and the transducer outcome for the source query (right) are presented.

Concerning provenance, we selected 25 queries from well-known benchmarks: FedBench [SGH+11b], SP2B [SHLP09], and QALD[22]. FedBench is a comprehensive benchmark suite used to test and analyze the performance of federated query processing strategies. FedBench recompiles 32 queries from heterogeneous life science, cross-domain, and linked-data sources. SP2B is a publicly available, language-specific SPARQL-performance benchmark that presents 12 queries in DBLP syntax and varying in general features, such as selectivity, query, and output size, and involving different types of joins. Finally, QALD is a series of evaluation campaigns concerning multilingual QA over linked data. From QALD, we considered QALD2, a training set of 100 natural-language questions with corresponding SPARQL queries and correct answers using DBpedia and MusicBrainz datasets, and QALD4, a question-training set comprising 25 questions over the biomedical datasets SIDER, Diseasesome, and Drugbank. From those benchmarks, we selected queries defined for some of the datasets listed in the subsection A.5.1.1 and having a different syntactic structure to previously selected ones. From Fedbench, we extracted queries numbered as follows: Q1, Q2, Q13, Q14, Q15, Q18, Q19, Q28, Q29, Q30, Q34, Q35, Q36, and Q44. From SP2B, we extracted queries numbered as follows: Q45, Q46, Q47, Q48, and Q50. From QALD2 and QALD4, we adapted queries numbered as follows: Q3, Q4, Q20, Q21, Q22, and Q23.

Furthermore, 25 additional queries were selected from the Linked Open Data SPARQL-endpoints logs (from the 2014 period), in particular from the selected datasets mentioned in the previous subsection A.5.1.1 (DBpedia, MusicBrainz, BNE, BNF, BNB, and Drugbank). Their distribution was the following: 10 queries from the Dbpedia log (Q5, Q6, Q7, Q8, Q9, Q10, Q16, Q17, Q32, and Q49), 2 queries from the MusicBrainz log (Q11 and Q12), 9 queries from the bibliographic domains BNE (Q31, Q33, and Q42), BNF (Q40, Q41, and Q43), and BNB logs (Q37, Q38, and Q39), and 4 queries from the DrugBank log (Q24, Q25, Q26, and Q27). To select these, clustering of the queries was performed according to their syntactic structure, and a random sample from each group was chosen, previously eliminating those queries that were malformed or did not meet our structure criteria.

Regarding syntactic structure, we ensured that a variety of the SPARQL operators (UNION, OPTIONAL, and FILTER) were present and different joins of variables appeared in the queries, as well as a different number of triple patterns (a minimum of 1 and a maximum of 7 triple patterns).

In principle, the natural way to assess the accuracy of the outcome query is to compare its answers against a set of expected answers. However, the set of expected answers can change along with the target dataset, because dataset content can differ (this is one reason to query a target dataset) or even change dynamically. Therefore, a compromise would be to synthesize a set of expected answers for each dataset based on a query expression that would obtain those expected answers. This criteria allows sustainment of the relevance of the gold standard over time and management of situations where the source query makes it difficult to anticipate

[22] http://nlp.uned.es/clef-qa/

the expected answers in a target dataset without evaluation of a proper query. For example, a query run on different weather-station datasets (reporting measures from different places) and asking for the number of days in 2016 registering a temperature less than 0^oC.

We used gold-standard queries as references to assess the accuracy of the outcome queries. A *Gold standard* of a query, Q_s, for a source dataset, D_s, is a query, Q_t, adequate for a target dataset, D_t, defined by a person who knows D_s and D_t and with the intention of being the most similar to Q_s for obtaining answers intended from the source query, but in the target dataset. We performed the following to obtain and validate these queries: upon selection of the source queries, members of our laboratory with expertise in querying the selected datasets were asked to write corresponding gold-standard responses to each selected source query; and afterward, each corresponding pair of queries was assessed and approved for evaluation.

A.5.2 Analysis of transducer outcomes

Given a query and a target dataset, the *transducer outcome* represents an adequate query for the target dataset that results from the application of the transformation rules (described in Section A.4) to the original query. To answer the questions posed at the beginning of this section, we ran the transducer using every original query in the benchmark, followed by use of the gold-standard and outcome queries over the corresponding target dataset for every original query. Each transducer outcome joined together with those for the gold-standard query is displayed in the tables in Appendix A.7.

We compared every pair of gold-standard and outcome queries with respect to their text and their results. To compare their results we calculated the following accuracy metrics, typical from the information retrieval domain. We call *Relevant answers* (*Rel*) to the set of answers obtained by running the gold-standard query, and *Retrieved answers* (*Ret*) to the set of answers obtained by running the outcome query. The measures Precision (P), Recall (R), and F-measure ($F1$) were calculated with the following formulae (see the results in tables A.5, A.6, and A.7).[23]

$$P = \frac{|Rel \cap Ret|}{|Ret|} \quad R = \frac{|Rel \cap Ret|}{|Rel|} \quad F1 = 2 \times \frac{P \times R}{P + R}$$

Analysis of the results revealed the following grouping of queries (see Table A.8). In 22 of 50 queries, the text of the outcome and gold-standard queries were equal. For 9 of these queries (Q13, Q14, Q16, Q19, Q20, Q21, Q31, Q38, and Q48), only equivalence rules were applied to obtain the outcome, and Q11 was already adequate for the target dataset. Interestingly, in the remaining 12 queries (Q3, Q6, Q8, Q10, Q12, Q25, Q26, Q30, Q33, Q36, Q42, and Q49), equivalence rules were not applied or were insufficient to obtain an adequate query.

In the remaining queries (28 of 50), the outcome text differed from that of the gold-standard expression. Two subgroups could be distinguished.

[23] The experiments were conducted on 2015-05-15.

TABLE A.5: Accuracy metrics for outcomes from queries for the media domain.

Domain:	Media-Domain								
Queries	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
|Rel|	12	5	9	20	358	39	6	4	3
|Ret|	12	4103	9	20	496	39	20	4	1
|Ret ∩ Rel|	12	5	9	20	358	39	6	4	1
P	1	0.001	1	1	0.72	1	0.3	1	1
R	1	1	1	1	1	1	1	1	0.33
F1	1	0.002	1	1	0.83	1	0.46	1	0.5
Queries	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	
|Rel|	100	0	1	4	1	1	19	0	
|Ret|	100	0	1	4	1	1	19	0	
|Ret ∩ Rel|	100	0	1	4	1	1	19	0	
P	1	1	1	1	1	1	1	1	
R	1	1	1	1	1	1	1	1	
F1	1	1	1	1	1	1	1	1	

TABLE A.6: Accuracy metrics for outcomes from queries for the life science domain

Domain:	LifeScience-Domain						
Queries	Q18	Q19	Q20	Q21	Q22	Q23	Q24
|Rel|	1	6	1	173	1	11	511
|Ret|	1	6	1	173	2	1	15
|Ret ∩ Rel|	1	6	1	173	1	1	15
P	1	1	1	1	0.5	1	1
R	1	1	1	1	1	0.09	0.02
F1	1	1	1	1	0.66	0.16	0.05
Queries	Q25	Q26	Q27	Q28	Q29	Q30	
|Rel|	0	0	147	10	2	100	
|Ret|	0	0	1	0	3	100	
|Ret ∩ Rel|	0	0	1	0	2	100	
P	1	1	1	1	0.66	1	
R	1	1	0.006	0	1	1	
F1	1	1	0.01	0	0.8	1	

TABLE A.7: Accuracy metrics for outcomes from queries for the bibliographic domain

Domain:	Bibliographic-Domain									
Queries	Q31	Q32	Q33	Q34	Q35	Q36	Q37	Q38	Q39	Q40
|Rel|	37	1	1	301	682	26	10	1	0	10
|Ret|	37	1	1	4	19827	26	1	1	0	0
|Ret ∩ Rel|	37	1	1	4	682	26	1	1	0	0
P	1	1	1	1	0.034	1	1	1	1	1
R	1	1	1	0.013	1	1	0.1	1	1	0
F1	1	1	1	0.026	0.066	1	0.18	1	1	0
Queries	Q41	Q42	Q43	Q44	Q45	Q46	Q47	Q48	Q49	Q50
|Rel|	0	1	10	3	1500	4	12000	51	13	1231
|Ret|	0	1	10	16	100	28	12000	51	13	578
|Ret ∩ Rel|	0	1	10	3	75	4	12000	51	13	415
P	1	1	1	0.18	0.75	0,14	1	1	1	0.71
R	1	1	1	1	0.05	1	1	1	1	0.33
F1	1	1	1	0.31	0.09	0.25	1	1	1	0.45

TABLE A.8: Classification summary of outcome queries.

Query Set 50 queries (100%)						
Outcome = Gold Standard 22 queries		Outcome ≠ Gold Standard 28 queries				
Ret = Rel 22 queries		Ret=Rel 10 queries		Ret ≠ Rel 18 queries		
Only E rules 10 queries	Not only E rules 12 queries	Only E rules 2 query	Not only E rules 8 queries	P=1 R<1 8 queries	P<1 R=1 8 queries	P<1 R<1 2 queries

One subgroup included 10 outcome queries that provided the same set of answers as the corresponding gold-standard queries (*Ret = Rel*). Query Q47 was already adequate for the target dataset, and the outcome for Q15 was achieved by applying only equivalence rules. The remaining 8 queries (Q1, Q4, Q17, Q18, Q32, Q39, Q41, and Q43) required different types of rules. The second subgroup included 18 outcome queries that provided different sets of answers than the corresponding gold-standard queries (*Ret ≠ Rel*), giving rise to three different cases: 1) queries (Q9, Q23, Q24, Q27, Q28, Q34, Q37, and Q40) that achieved full precision, but suffered a loss in recall; 2) queries (Q2, Q5, Q7, Q22, Q29, Q35, Q44, and Q46) that achieved full recall, but suffered a loss in precision; and 3) queries (Q45 and Q50) that suffered a loss in recall and precision.

In summary, in 32 of 50 source queries, the outcome queries provided the same set of answers as the corresponding gold-standard queries. In particular, 6 of these (Q11, Q17, Q25, Q26, Q39, and Q41) resulted in empty sets of relevant and retrieved answers.

Table A.9 presents the queries of the benchmark grouped according to the types of rules actually applied by the transducer to obtain the corresponding outcome. Notably, each rule appeared relevant at different

TABLE A.9: Distribution of queries according to the types of rules applied.

Packet of rules	Number of queries	List of queries
No rules	3	Q11, Q45, Q47
E	10	Q13, Q14, Q15, Q16, Q19, Q20, Q21, Q31, Q38, Q48
E+H	1	Q4
H	1	Q50
E+H+F	4	Q10, Q17, Q28, Q32
E+A	1	Q39
A	2	Q5, Q6
A+P	1	Q42
P	1	Q3
E+P	10	Q1, Q7, Q8, Q9, Q12, Q22, Q23, Q33, Q34, Q36
E+P+F	1	Q2
P+F	2	Q18, Q40
E+F	8	Q24, Q25, Q26, Q27, Q30, Q44, Q46, Q49
F	5	Q29, Q35, Q37, Q41, Q43

moments, although the equivalence (E), profile-based (P), and feature-based (F) rules appeared more frequently applied. Specifically, E rules were applied on 35 occasions, hierarchy (H) rules were applied on 6 occasions, answer-based (A) rules were applied on 4 occasions, P rules were applied on 15 occasions, and F rules were applied on 20 occasions.

For queries Q11, Q45, and Q47, no rules were necessary, because their terms were already adequate for the target dataset. Although A rules appeared to be infrequently applied, it is notable that the sole application succeeded for two of the queries outside of collaborating in the transformation of two additional queries. Therefore, this suggested that A rules should not be discarded, because their contribution can be significant under certain situations. Moreover, we observed that some pairs of rule types did not collaborate to obtain transformations of the queries in the benchmark. Specifically, pairs (H, A), (H, P), and (A, F) never worked in a joint package, (A, P) only worked in one query, and (P, F) worked in two queries. However, these results were incidental, because none of the rules structurally precluded the application of any other, with the results dependent upon the context and existing term mappings.

A.5.3 Processing time

For each query in the benchmark, Table A.17 in Appendix A.8 displays the following time values (in *ms*). Column TT displays the time required by the transducer to obtain the outcome. Column AT displays the time needed to obtain answers to the queries in the corresponding SPARQL endpoint. Column TAT displays the sum of the previous times (TT+AT). Column GSAT displays the time required to obtain an answer to the gold-standard query in the corresponding SPARQL endpoint. Column AT−GSAT shows the differences between times expressed by both columns. Columns E, H, A, P, and F, respectively, display the times required by the transducer to apply the different rules. Figures A.6 and A.7

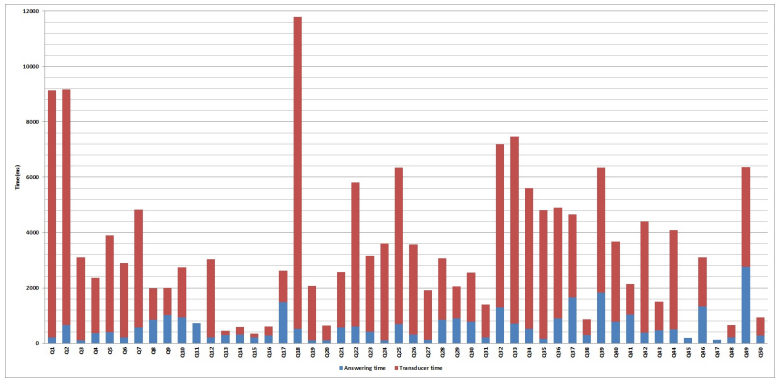


FIGURE A.6: Answering time plus Transducer time.

graphically display the same information (except for AT−GSAT). Column Rule[Num] in Table A.17 displays the rules applied and the number of times for each.

In an uncontrolled environment, such as like the Web of Data, the running times of queries vary greatly. For that reason, we performed our tests five times, with Figures A.6 and A.7 showing the averages of those execution times. We also performed warm-up queries to avoid initial delays.

Total times (TAT) ranged from a lower bound of 122*ms* to an upper bound of 11787*ms*. The time dedicated to the transducer (TT) ranged from a lower bound of 0*ms* to an upper bound of 11264*ms*. Nevertheless, every query, except for Q1, Q2, and Q18, was transformed in less than 6749*ms*.

Regarding the execution times of the outcome queries (AT) versus the gold-standard queries (GSAT), we observed that the differences in times (see column AT−GSAT) was not always uniform, with the outcome query outperforming the gold-standard query in some instances (by a maximum value of 396*ms* in Q33) and vice versa (by a maximum value of 377*ms* in Q17). In summary, our results indicated that in 30 of the queries, the execution time of the outcome queries was higher, with an average of 89.1*ms*, whereas in the other 20 instances, it the gold standard had a higher execution time, with an average value of 84.15*ms*. These results suggested that although the time differences can be considered insignificant, the tests showed that the transducer incurred a small surcharge in execution time. The main reason for this overload was the introduction of two new features by the rewriting process: UNION clauses, introduced when applying hierarchy rules (as in Q4, with a time surcharge of 15*ms*) or answer-based rules (as in Q5, with a time surcharge of 68*ms*), and query generalization due to feature-rule application (as in Q17, with 377*ms*, or in Q32, with 295*ms*).

We considered these processing times encouraging, because the prototype implementation can be optimized. Therefore, the fourth question regarding the acceptability of the processing time can be answered in the affirmative.

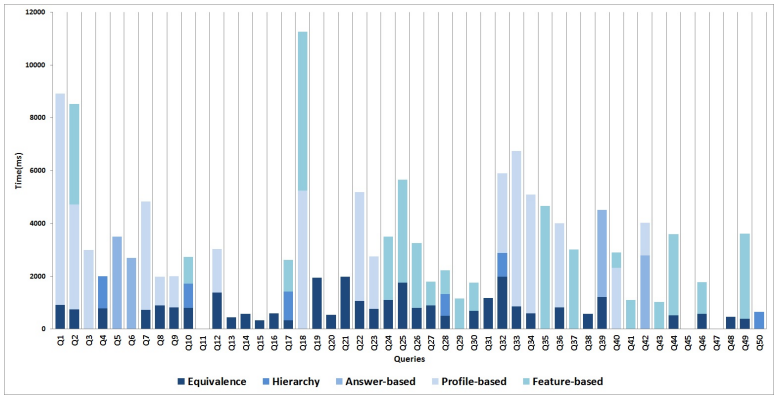


FIGURE A.7: Transducer time grouped by kinds of rules.

A.5.4 Final discussion

Based on our observations, we were encouraged by the results showing that 35 of 50 source queries of the benchmark needed more than equivalence and hierarchy rules to be transformed to an adequate query (see the queries with a value in the A, P, or F columns in Table A.17 in Appendix A.8). Therefore, translation techniques constrained to semantics-preserving transformations or only accounting for hierarchical relaxation would not be able to satisfactorily process such queries. Therefore, those systems would return nothing to the user in such frequent cases. By contrast, our rewriting system offered an acceptable query transformation, answering the first question of whether other rules different from equivalence and hierarchy rules are relevant in the affirmative.

With respect to the second question, (Is the transformation process relevant enough to be maintained in its entirety without skipping any kind of rule?), we found that some rules were more frequently applied (equivalence, profile-based, and feature-based rules, specifically). However, hierarchy and answer-based rules also contributed significantly under certain situations, and their removal would likely decrease the overall quality of the approach. Given to the nature of the rules and the order in which they were applied, we are unaware of a statistically robust method of evaluating each rule in isolation. We believe that application of each rule type in isolation to the benchmark would likely produce disappointing results.

Furthermore, none of the rules imposed a high processing time; therefore, the benefit/cost ratio for their use is clearly positive (see Table A.17 in Appendix A.8). Furthermore, discarding any of them would likely hinder the overall performance of the approach. Consequently, we believe that the second question posed at the beginning of this section can be answered in the affirmative.

In summary, 64% of the outcome queries of the benchmark obtained the same set of answers as the gold-standard queries, with an additional 16% of the outcome queries preserving full recall of the expected answers,

although they did not achieve full precision, and another 16% of the outcome queries preserving full precision with respect the expected answers, but not achieving complete recall. The remaining 4% of the outcome queries did not achieve full precision or complete recall (see Table A.8 for a graphical display of the results).

Consequently, with respect to the third question, the accuracy results were acceptable, and the transformation rules implemented in the transducer deserve merit. Our findings support additional research into further implementation of this technique.

The prototype of the presented rewriting system is implemented in Java and uses the AGG framework[24] for rule processing. An implementation for demonstration purposes is available online[25].

A.6 Conclusion

Because the use of Linked Open Datasets to extract actionable information has become common, the development of systems providing easy access and navigation of such datasets has acquired increased relevance. In this study, we developed a method that allows users to query distributed and heterogeneous datasets using only the vocabulary with which they are familiar. A transducer was developed to transform the original query expressions into queries adapted to the vocabulary of the target dataset.

Additionally, the rewriting rules used by the transducer were presented. In addition to equivalence, hierarchy, and EDOAL alignments encountered among datasets (considered by a majority of query processing approaches), the transducer manages other kind of rules that enhanced the likelihood of obtaining new query expressions. Therefore, one novel contribution of the proposal is the definition and implementation of three new types of rules: answer-based, profile-based, and feature-based. These rules manage three different types of context information in regard to the optimal query outcome from the target dataset. The answer-based rules use bindings of variables in the original query during query performance on the source dataset, the profile-based rules use triples in the source dataset in regard to each inadequate IRI, and the feature-based rules generalize inadequate IRIs by variables while profiling those variables according to a description of the intended resource. These rules allow the capture of semantics that are relevant and not explicitly expressed in the existing alignments. The exploitation of the transducer outcomes will improve user experience relative to other methods that only apply semantics-preserving transformations and do nothing in the event that such a goal is unachievable.

Furthermore, to evaluate the feasibility of our approach we constructed a proper benchmark for the experiments, because we were unable to locate suitable benchmarks specific for our requirements. Our results were assessed according as follows: metrics (precision, recall, and F-measure) that reflected the quality of the answer set of the queries obtained by the

[24] http://www.user.tu-berlin.de/o.runge/agg/
[25] http://sml.tecnalia.com/SparqlRewriting/faces/demoUser.xhtml

transducer with respect to the answers expected, the relevance of each kind of rule, and the execution time of those queries with result to transformation and running time.

Our future work involves development of a mechanism allowing estimation of query rewriting quality over the Linked Open Data.

A.7 Benchmark

TABLE A.10: Query set 1

Query 1(FedBench CD4) - LinkedMDB	
SELECT ?actor WHERE ?film purlc:title 'Top Gun' . ?film movie:actor ?actor .	
Gold Standard - DBpedia: SELECT ?actor WHERE ?film foaf:name 'Top Gun' . ?film db-o:starring ?actor .	**Transducer outcome** - DBpedia: SELECT ?actor WHERE { ?film foaf:name 'Top Gun' .} UNION { ?film rdfs:label 'Top Gun' .} ?film db-o:starring ?actor .
Query 2(FedBench CD5) - LinkedMDB	
SELECT ?film ?director ?genre WHERE ?film movie:director ?director. ?film movie:country movie:country/IT. ?film movie:genre ?genre .	
Gold Standard - DBpedia SELECT ?film ?director ?genre WHERE ?film db-o:director ?director. ?film db-p:country 'Italy'. ?film db-o:genre ?genre .	**Transducer outcome** - DBpedia SELECT ?film ?director ?genre WHERE ?film db-o:director ?director. ?film db-p:country db-r:Italy. ?film ?p ?genre .
Query 3(QALD4) - MusicBrainz	
SELECT ?name ?band WHERE ?artist rdfs:label ?name. ?artist mo:member_of ?band. ?band rdfs:label 'The Beatles' .	
Gold Standard - DBpedia SELECT ?name ?band WHERE ?artist foaf:name ?name.?band db-o:formerBandMember ?artist. ?band foaf:name 'The Beatles'.	**Transducer outcome** - DBpedia SELECT ?name ?band WHERE ?artist foaf:name ?name.?band db-o:formerBandMember ?artist. ?band foaf:name 'The Beatles'.
Query 4(QALD4) - MusicBrainz	
SELECT ?album WHERE ?album rdf:type mo:Record . ?album foaf:maker ?artist . ?artist foaf:name 'Slayer'.	
Gold Standard - DBpedia SELECT ?album WHERE ?album rdf:type db-o:Album . ?album db-o:artist ?artist. artist foaf:name 'Slayer'.	**Transducer outcome** - DBpedia SELECT ?album WHERE { ?album rdf:type db-o:Album . } union{ ?album rdf:type db-o:Single.} ?album db-o:artist ?artist. artist foaf:name 'Slayer'.
Query 5(DBPEDIA log) - DBpedia	
SELECT DISTINCT ?a WHERE ?a a db-o:Artist. ?a foaf:name ?n. FILTER (regex(?n, 'John','i')) .	
Gold Standard - LinkedMDB SELECT DISTINCT ?a WHERE ?a a foaf:Person . ?a rdfs:label ?n . FILTER (regex(?n, 'John','i')) .	**Transducer outcome** - LinkedMDB SELECT DISTINCT ?a WHERE { ?a a movie:editor .} UNION { ?a a movie:producer . } UNION { ?a a foaf:Person . } ?a rdfs:label ?n . FILTER (regex(?n, 'John','i')) .
Query 6(DBPEDIA log) - DBpedia	
SELECT DISTINCT ?a WHERE db-r:The_Other_Side_of_the_Wind ?p ?a .	
Gold Standard - LinkedMDB SELECT DISTINCT ?a WHERE movie:46921 ?p ?a .	**Transducer outcome** - LinkedMDB SELECT DISTINCT ?a WHERE movie:46921 ?p ?a .
Query 7(DBPedia Log) - LinkedMDB	
SELECT ?movieName WHERE { ?woody movie:director_name "Woody Allen". ?actor movie:actor_name "Julie Kavner". ?movie movie:director ?woody; movie:actor ?actor; purlc:title ?movieName.}	
Gold Standard - DBpedia SELECT ?movieName WHERE ?woody foaf:name "Woody Allen"@en. . ?actor foaf:name "Julie Kavner"@en. ?movie db-o:director ?woody; db-o:starring ?actor; db-p:title ?movieName. }	**Transducer outcome** - DBpedia SELECT ?movieName WHERE ?woody foaf:name "Woody Allen"@en. ?actor foaf:name "Julie Kavner"@en. ?movie db-o:director ?woody; db-o:starring ?actor; ?p ?movieName. }

TABLE A.11: Query set 2

Query 8(DBPedia Log) - LinkedMDB	
SELECT ?actor WHERE { movie-r:film/154 movie:actor ?actor. }	
Gold Standard - DBPedia SELECT ?actor WHERE db-r:E.T._the_ExtraTerrestrial db-o:starring ?actor . }	**Transducer outcome** - DBPedia SELECT ?actor WHERE db-r:E.T._the_ExtraTerrestrial db-o:starring ?actor . }
Query 9(DBPedia Log) - LinkedMDB	
SELECT ?writer WHERE { movie-r:film/8304 movie:writer ?writer . }	
Gold Standard - DBPedia SELECT ?writer WHERE { db-r:Interstellar_(film) db-o:writer ?writer . }	**Transducer outcome** - DBPedia SELECT ?writer WHERE { db-r-fr:Interstellar_(film) db-o:writer ?writer . }
Query 10(DBpedia log) - DBpedia	
SELECT distinct ?actor ?place WHERE {?actor a db-o:VoiceActor.?actor db-o:birthPlace ?place } LIMIT 100	
Gold Standard - LinkedMDB SELECT distinct ?actor ?place WHERE {?actor a movie:actor. ?actor ?p ?place . } LIMIT 100	**Transducer outcome** - LinkedMDB SELECT distinct ?actor ?place WHERE {?actor a movie:actor. ?actor ?p ?place . } LIMIT 100
Query 11(MusicBrainz log) - MusicBrainz	
SELECT ?uri ?name ?label WHERE { ?uri a mo:MusicArtist; foaf:name ?name; rdfs:label ?label. FILTER (?name = "The Beatles"). FILTER (lang(?label) = "en"). }	
Gold Standard - Jamendo SELECT ?uri ?name ?label WHERE { ?uri a mo:MusicArtist; foaf:name ?name; rdfs:label ?label. FILTER(?name = "The Beatles"). FILTER (lang(?label) = "en")}	**Transducer outcome** - Jamendo SELECT ?uri ?name ?label WHERE { ?uri a mo:MusicArtist; foaf:name ?name; rdfs:label ?label. FILTER(?name = "The Beatles"). FILTER (lang(?label) = "en")}
Query 12(MusicBrainz log) - MusicBrainz	
SELECT (STR(?title) AS ?stitle) ?album WHERE { ?a foaf:made ?album. ?album a mo:SignalGroup; dc:title ?title. } order by ?title	
Gold Standard - DBpedia SELECT (STR(?title) as ?stitle) ?album WHERE { foaf:made ?album.?album a db-o:MusicalWork; dc:title ?title} order by ?title	**Transducer outcome** - DBpedia SELECT (STR(?title) as ?stitle) ?album WHERE { foaf:made ?album.?album a db-o:MusicalWork; dc:title ?title} order by ?title
Query 13(Fedbench) - DBpedia	
SELECT ?predicate ?object WHERE { dbpedia:Barack_Obama ?predicate ?object . }	
Gold Standard - New York Times SELECT ?predicate ?object WHERE { data-nyt:47452218948077706853 ?predicate ?object . }	**Transducer outcome** - New York Times SELECT ?predicate ?object WHERE { data-nyt:47452218948077706853 ?predicate ?object . }
Query 14(Fedbench) - DBpedia	
SELECT ?name ?location WHERE { ?artist foaf:based_near dbpedia:Baden-Baden . ?location geonames:parentFeature ?germany . }	
Gold Standard - New York Times SELECT ?name ?location WHERE { ?artist foaf:based_near data-nyt:N17957130964238223481. ?location geonames:parentFeature ?germany . }	**Transducer outcome** - New York Times SELECT ?name ?location WHERE { ?artist foaf:based_near data-nyt:N17957130964238223481. ?location geonames:parentFeature ?germany . }
Query 15(Fedbench) - DBpedia	
SELECT ?name WHERE { dbpedia:California geonames:name ?name ; foaf:isPrimaryTopicOf ?news }	
Gold Standard - New York Times SELECT ?name WHERE { data-nyt:56975960096322330571 geonames:name ?name ; nytimes:topicPage ?news }	**Transducer outcome** - New York Times SELECT ?name WHERE { data-nyt:56975960096322330571 geonames:name ?name ; foaf:isPrimaryTopicOf ?news }

TABLE A.12: Query set 3

Query 16 (Dbpedia log) - DBPedia	
SELECT ?y WHERE { ?y ?p dbpedia:Fernando_Alonso }	
Gold Standard - BBC SELECT ?y WHERE { ?y ?p bbc-r:f63149e9-c577-40b4-bd88-1efac54228db }	**Transducer outcome** - BBC SELECT ?y WHERE { ?y ?p bbc-r:f63149e9-c577-40b4-bd88-1efac54228db }
Query 17(Dbpedia log) - Dbpedia	
SELECT ?city WHERE { dbpedia: Seattle_80_Tacoma_International_Airport a dbp-owl : Airport ; dbp-owl : city ?city . }	
Gold Standard - BBC SELECT ?city WHERE { bbc-r:d3cbd167-06d7-42c3-b23a-cce2657356c8#id a geonames:Place; ?p ?city . }	**Transducer outcome** - BBC SELECT ?city WHERE { bbc-r:d3cbd167-06d7-42c3-b23a-cce2657356c8#id a owl:Thing; ?p ?city .}
Query 18(FedBench LS1) - Drugbank	
SELECT ?drug ?melt WHERE ?drug rdf:type drugbank:drugs . ?drug drugbank:meltingPoint ?melt .	
Gold Standard - DBPedia SELECT ?drug ?melt WHERE ?drug rdf:type db-o:drug . ?drug db-o:meltingPoint ?melt.	**Transducer outcome** - DBPedia SELECT ?drug ?melt WHERE ?drug rdf:type db-o:drug . ?drug ?p ?melt.
Query 19(FedBench LS2) - Drugbank	
SELECT ?predicate ?object WHERE drugbank-drugs:DB00201 ?predicate ?object .	
Gold Standard - KEGG SELECT ?predicate ?object WHERE kegg:D005281 ?predicate ?object .	**Transducer outcome** - KEGG SELECT ?predicate ?object WHERE kegg:D005281 ?predicate ?object .
Query 20(QALD4) - Drugbank	
SELECT ?x WHERE drugbank-drugs:DB00404 rdfs:label ?x .	
Gold Standard - SIDER SELECT ?x WHERE sider:2118 rdfs:label ?x .	**Transducer outcome** - SIDER SELECT ?x WHERE sider:2118 rdfs:label ?x .
Query 21(QALD4) - SIDER	
SELECT ?p1 ?y1 ?p2 ?y2 WHERE sider:1690 ?p1 ?y1 . sider:119607 ?p2 ?y2 .	
Gold Standard - Drugbank SELECT ?p1 ?y1 ?p2 ?y2 WHERE drugbank-drugs:: DB00445 ?p1 ?y1 . drugbank-drugs:: DB00580 ?p2 ?y2 .	**Transducer outcome** - Drugbank SELECT ?p1 ?y1 ?p2 ?y2 WHERE drugbank-drugs:: DB00445 ?p1 ?y1 . drugbank-drugs:: DB00580 ?p2 ?y2 .
Query 22(QALD4) - Drugbank	
SELECT DISTINCT ?v0 ?v1 WHERE drugbank-drugs:DB00194 drugbank:molecularWeightAverage ?v0. OPTIONAL { drugbank-drugs:DB00194 drugbank: molecularWeightMono ?v1. } }	
Gold Standard - DBPedia SELECT DISTINCT ?v0 WHERE dbpedia-r:Vidarabine db-o:molecularWeight ?v0 .	**Transducer outcome** - DBPedia SELECT DISTINCT ?v0 ?v1 WHERE db-r:Vidarabine db-o:molecularWeight ?v0 . OPTIONAL {db-r:Vidarabine db-o:molecularWeight ?v1. } .
Query 23(QALD4) - SIDER	
SELECT DISTINCT ?x WHERE sider:8378 sider:drugName ?x	
Gold Standard - DBPedia SELECT DISTINCT ?x WHERE db-r:Allopurinol rdfs:label ?x.	**Transducer outcome** - DBPedia SELECT DISTINCT ?x WHERE db-r:Allopurinol db-o:tradename ?x.

TABLE A.13: Query set 4

Query 24(Drugbank log) - Drugbank	
SELECT DISTINCT ?dr ?br ?disease WHERE { drugbank-r:drugs/DB00115 drugbank:possibleDiseaseTarget ?disease; rdfs:label ?dr ; drugbank:brandName ?br.}	
Gold Standard - DISEASOME SELECT DISTINCT ?dr ?br ?disease WHERE { ?disease diseasome:possibleDrug drugbank-d:DB00115; rdfs:label ?dr; diseasome:name ?br . }	**Transducer outcome** - DISEASOME SELECT DISTINCT ?dr ?br ?disease WHERE { drugbank-d:DB00115 ?p ?disease ; rdfs:label ?dr ; ?p2 ?br . }
Query 25(Drugbank log) - Drugbank	
SELECT ?IntEffect WHERE { ?Int drugbank:interactionDrug1 drugbank-d:DB01203 . ?Int drugbank:interactionDrug2 drugbank-d:DB00414 . ?Int drugbank:text ?IntEffect .}	
Gold Standard - DBpedia SELECT ?IntEffect WHERE { ?Int ?a db-r:Nadolol. ?Int ?b db-r:Acetohexamide. ?Int ?c ?IntEffect .}	**Transducer outcome** - DBpedia SELECT ?IntEffect WHERE { ?Int ?a db-r:Nadolol. ?Int ?b db-r:Acetohexamide. ?Int ?c ?IntEffect .}
Query 26(Drugbank log) - Sider	
SELECT ?drug WHERE { ?sider sider:drug_name ?drug . ?sider sider:side_effect ?effect . FILTER regex (?drug,"Fenofibrate|Aspirin|Rosuvastatin|Levothyroxine|Valsartan","i") }	
Gold Standard - Drugbank SELECT ?drug WHERE { ?sider drugbank:drug_name ?drug . ?sider ?p ?effect . FILTER regex (?drug,"Fenofibrate|Aspirin| } Rosuvastatin|Levothyroxine|Valsartan","i") . }	**Transducer outcome** - Drugbank SELECT ?drug WHERE{ ?sider drugbank:drug_name ?drug . ?sider ?p ?effect . FILTER regex (?drug,"Fenofibrate|Aspirin| Rosuvastatin|Levothyroxine|Valsartan","i") . }
Query 27(Drugbank log) - SIDER	
SELECT ?se ?sen WHERE {sider:drugs/4095 sider:sideEffect ?se .?se sider:sideEffectName ?sen .}	
Gold Standard - Drugbank SELECT ?se WHERE { drugbank-d::DB00333 drugbank:toxicity ?se.}	**Transducer outcome** - Drugbank SELECT ?se ?sen WHERE { drugbank-d:DB00333 ?p1 ?se. ?se ?p2 ?sen.}
Query 28(FedBench) - KEEG	
SELECT ?drugDesc ?cpd WHERE { ?drug drugbank:keggCompoundId ?cpd . ?drug rdfs:label ?drugDesc . ?enzyme kegg:xSubstrate ?cpd . ?enzyme rdf:type kegg:Enzyme .}	
Gold Standard - Drugbank SELECT ?drugDesc ?cpd WHERE { ?drug drugbank:keggCompoundId ?cpd . ?drug drugbank:description ?drugDesc . ?enzyme ?p1 ?cpd . ?enzyme rdf:type ?o1 .}	**Transducer outcome** - Drugbank SELECT ?drugDesc ?cpd WHERE { { ?drug drugbank:keggCompoundId ?cpd . ?drug rdfs:label ?drugDesc . ?enzyme ?p1 ?cpd . ?enzyme rdf:type drugbank:drugs. }
Query 29(FedBench) - CHEBI	
SELECT ?drug ?chebiImage WHERE { ?drug foaf:name ?drugBankName . ?chebiDrug purl:title ?drugBankName . ?chebiDrug chebi:image ?chebiImage . }	
Gold Standard - Drugbank SELECT ?drug ?chebiImage WHERE { ?drug drugbank:genericName ?drugBankName . ?chebiDrug purl:title ?drugBankName . ?chebiDrug ?p1 ?chebiImage . }	**Transducer outcome** - Drugbank SELECT ?drug ?chebiImage WHERE { ?drug foaf:name ?drugBankName . ?chebiDrug purl:title ?drugBankName . ?chebiDrug ?p1 ?chebiImage . }
Query 30(FedBench) - Drugbank	
SELECT ?drug ?keggDrug WHERE { ?drug drugbank:drugCategory drugbank-category:micronutrient . ?drug drugbank:casRegistryNumber ?id . ?keggDrug bio2rdf:xRef ?id . } LIMIT 100	
Gold Standard - KEGG SELECT ?drug ?keggDrug WHERE { ?drug ?p1 ?o1 . ?drug ?p2 ?id . ?keggDrug bio2rdf:xRef ?id . } LIMIT 100	**Transducer outcome** - KEGG SELECT ?drug ?keggDrug WHERE { ?drug ?p1 ?o1 . ?drug ?p2 ?id . ?keggDrug bio2rdf:xRef ?id . } LIMIT 100

TABLE A.14: Query set 5

Query 31(BNE log) - BNE	
SELECT DISTINCT ?p ?o WHERE bne-resource:XX1718747 ?p ?o .	
Gold Standard - BNB	**Transducer outcome** - BNB
SELECT DISTINCT ?p ?o WHERE bnb:CervantesSaavedraMiguelde1547-1616 ?p ?o .	SELECT DISTINCT ?p ?o WHERE bnb:CervantesSaavedraMiguelde1547-1616 ?p ?o .
Query 32(DBpedia log) - DBpedia	
SELECT DISTINCT ?label ?author WHERE db-r:Pride_and_Prejudice a db-o:Book . db-r:Pride_and_Prejudice foaf:name ?label . db-r:Pride_and_Prejudice db-o:author ?author . ?author a db:owl:Artist .	
Gold Standard - Cambridge	**Transducer outcome** - Cambridge
SELECT DISTINCT ?label ?author WHERE cam:cambrdgedb_...79f80074f a owl:Thing . cam:cambrdgedb_...79f80074f rdfs:label ?label . cam:cambrdgedb_...79f80074f dct:creator ?author . ?author a foaf:Person .	SELECT DISTINCT ?label ?author WHERE cam:cambrdgedb_...79f80074f a ?o . cam:cambrdgedb_...79f80074f rdfs:label ?label . cam:cambrdgedb_...79f80074f dct:creator ?author . ?author a foaf:Person.
Query 33(BNE log) - DBpedia	
SELECT ?a ?birth WHERE ?a db-o:writer db-r:Leo_Tolstoy . OPTIONAL { db-r:Leo_Tolstoy db-owl:dateOfBirth ?birth. }	
Gold Standard - BNE	**Transducer outcome** - BNE
SELECT ?a ?birth WHERE bne-r:XX933715 fr:P2010 ?a . OPTIONAL { bne-r:XX933715 fr:P3040 ?birth. }.	SELECT ?a ?birth WHERE bne-r:XX933715 fr:P2010 ?a . OPTIONAL { bne-r:XX933715 fr:P3040 ?birth. }.
Query 34(Fedbench) - DBLP	
SELECT * WHERE ?a akt:has-author dblp-r:person/100007. OPTIONAL {?e akt:edited-by dblp-r:person/100007. }	
Gold Standard - DBpedia	**Transducer outcome** - DBpedia
SELECT * WHERE ?a db-o:author db-r:Tim_Berners-Lee . OPTIONAL {?e ?p db-r:Tim_Berners-Lee .}	SELECT * WHERE ?a db-o:author db-r:Tim_Berners-Lee . OPTIONAL {?e db-o:owner db-r:Tim_Berners-Lee .}
Query 35(Fedbench) - DBpedia	
SELECT DISTINCT ?a ?r WHERE ?a db-o:publishedIn ?r	
Gold Standard - DBLP	**Transducer outcome** - DBLP
SELECT DISTINCT ?a ?r WHERE ?a dblp:article-of-journal ?r	SELECT DISTINCT ?a ?r WHERE ?a ?p ?r
Query 36(Fedbench) - DBLP	
SELECT ?article WHERE ?article dblp:Article_IsWrittenBy ?author . ?author akt:full-name ?Author . FILTER (regex (?Author, 'Tanenbaum', 'i')) ORDER BY ?Author	
Gold Standard - BNB	**Transducer outcome** - BNB
SELECT ?article WHERE ?author bnb:hascreated ?article . ?author foaf:name ?Author . FILTER (regex(?Author, 'Tanenbaum', 'i')) ORDER BY ?Author	SELECT ?article WHERE ?author bnb:hascreated ?article . ?author foaf:name ?Author . FILTER (regex(?Author, 'Tanenbaum', 'i')) ORDER BY ?Author
Query 37(BNB log) - BNB	
SELECT ?book ?isbn ?title WHERE { ?book dct:creator bnb-person:Althea; bibo:isbn10 ?isbn; dct:title ?title. } LIMIT 10	
Gold Standard - BNF	**Transducer outcome** - BNF
SSELECT ?book ?isbn ?title WHERE { ?book dct:creator bnf-ark:12148/cb12130; bibo:isbn10 ?isbn; dct:title ?title. } LIMIT 10	SELECT ?book ?isbn ?title WHERE { ?book dct:creator ?author; bibo:isbn10 ?isbn; dct:title ?title.} LIMIT 10

TABLE A.15: Query set 6

Query 38(BNB log) - BNB	
SELECT * WHERE { ?subject ?predicate bnb-person:AustenJane1775-1817}	
Gold Standard - BNF SELECT * WHERE {?subject ?predicate <http://catalogue.bnf.fr/ark:/12148/cb118896036> }	**Transducer outcome** - BNF SELECT * WHERE {?subject ?predicate <http://catalogue.bnf.fr/ark:/12148/cb118896036> }
Query 39(BNB log) - BNB	
SELECT DISTINCT ?book ?title WHERE { ?book bibo:isbn10 '0851221033'. ?book a bibo:Book . ?book dct:title ?title.}	
Gold Standard - DBpedia SELECT DISTINCT ?book ?title WHERE { ?book db-o:isbn '0851221033'. ?book a dbpedia-owl:Book . ?book rdfs:label ?title.}	**Transducer outcome** - DBpedia SELECT DISTINCT ?book ?title WHERE { ?book db-o:isbn '0851221033'. ?book a dbpedia-owl:Book . ?book dct:title ?title.}
Query 40(BNF log) - BNF	
SELECT ?auteur ?jour ?nom WHERE { ?auteur foaf:birthday ?jour. ?doc bnf:r220 ?auteur. OPTIONAL {?auteur foaf:name ?nom } } ORDER BY (?jour) LIMIT 10	
Gold Standard - DBpedia SELECT ?auteur ?jour ?nom WHERE { ?auteur db-o:abstract ?jour. ?doc db-o:author ?auteur. OPTIONAL {?auteur foaf:name ?nom} } ORDER BY (?jour) LIMIT 10	**Transducer outcome** - DBpedia SELECT ?auteur ?jour ?nom WHERE { ?auteur ?p ?jour. ?doc db-o:author ?auteur. OPTIONAL {?auteur foaf:name ?nom} } ORDER BY (?jour) LIMIT 10
Query 41(BNF log) - BNF	
SELECT DISTINCT ?work ?title WHERE { ?work foaf:focus ?person. ?work dcterms:creator ?person ; rdfs:label ?title . } LIMIT 10	
Gold Standard - DBpedia SELECT DISTINCT ?work ?title WHERE { ?work foaf:focus ?person. ?work dcterms:creator ?person ; rdfs:label ?title . } LIMIT 10	**Transducer outcome** - DBpedia SELECT DISTINCT ?work ?title WHERE { ?work ?p1 ?person. ?work dcterms:creator ?person ; rdfs:label ?title . } LIMIT 10
Query 42(BNE log) - BNE	
SELECT DISTINCT ?item WHERE { ?item rdf:type frbr:C1003 . ?item isbd:P1004 "Decamerone" }	
Gold Standard - DBpedia SELECT DISTINCT ?item WHERE { ?item a db-o:Book. ?item db-o:titleOrig "Decamerone"@en . }	**Transducer outcome** - DBpedia SELECT DISTINCT ?item WHERE { ?item a db-o:Book. ?item db-o:titleOrig "Decamerone"@en . }
Query 43(BNF log) - BNF	
SELECT DISTINCT ?Edition WHERE { ?Edition a purlc:Manifestation; dcterms:subject ?subject. } LIMIT 10	
Gold Standard - DBpedia SELECT DISTINCT ?Edition WHERE { ?Edition a db-o:Work; dcterms:subject ?subject. } LIMIT 10	**Transducer outcome** - DBpedia SELECT DISTINCT ?Edition WHERE { ?Edition a ?o; ?p ?subject. } LIMIT 10
Query 44(Fedbench) - BNE	
SELECT DISTINCT ?item WHERE { ?item rdf:type frbr:C1002. ?item frbr:P2002 <http://datos.bne.es/resource/XX3383594> }	
Gold Standard - BNF SELECT DISTINCT ?item WHERE { ?item rdf:type ?i . ?item ?p bnf-ark:12148/cb131654249 }	**Transducer outcome** - BNF SELECT DISTINCT ?item WHERE { ?item rdf:type ?i . ?item ?p bnf:131654249 }

TABLE A.16: Query set 7

Query 45(SP2B - Q6) - SP2B

```
SELECT ?yr ?name ?document WHERE { ?class rdfs:subClassOf foaf:Document .
?document rdf:type ?class . ?document dc:creator ?author . ?author foaf:name ?name
OPTIONAL {?class2 rdfs:subClassOf foaf:Document . ?document2 rdf:type ?class2 .
?document2 dc:creator ?author2 .FILTER (?author=?author2) }}
```

Gold Standard - DBpedia

```
SELECT ?yr ?name ?document WHERE {
?class rdfs:subClassOf dbpedia:Work .
?document rdf:type ?class .
?document dc:creator ?author .
?author foaf:name ?name
OPTIONAL {
?class2 rdfs:subClassOf dbpedia:Work .
?document2 rdf:type ?class2 .
?document2 dc:creator ?author2 .
FILTER (?author=?author2 ) }}
```

Transducer outcome - DBpedia

```
SELECT ?yr ?name ?document WHERE {
?class rdfs:subClassOf foaf:Document .
?document rdf:type ?class .
?document dc:creator ?author .
?author foaf:name ?name
OPTIONAL {
O?class2 rdfs:subClassOf foaf:Document .
?document2 rdf:type ?class2 .
?document2 dc:creator ?author2 .
FILTER (?author=?author2 ) }}
```

Query 46(SP2B - Q12b) - SP2B

```
SELECT ?s WHERE { person:Paul_Erdoes sp:advisor ?s.
OPTIONAL{ person:John_von_Neumann sp:advisor ?s}}
```

Gold Standard - DBpedia

```
SELECT ?s WHERE
dbpedia:Paul_Erdoes db-owl:doctoralAdvisor ?s.
OPTIONAL{ dbpedia:John_von_Neumann
db-owl:doctoralAdvisor ?s } }
```

Transducer outcome - DBpedia

```
SELECT ?s WHERE
dbpedia:Paul_Erdoes ?p1 ?s.
OPTIONAL{ dbpedia:John_von_Neumann
?p2 ?s } }
```

Query 47(SP2B - Q9) - SP2B

```
SELECT DISTINCT ?predicate WHERE {
{ ?person rdf:type foaf:Person . ?subject ?predicate ?person}
UNION { ?person rdf:type foaf:Person . ?person ?predicate ?object }}
```

Gold Standard - DBpedia

```
SELECT DISTINCT ?predicate WHERE { {
?person rdf:type dbpedia-owl:Person .
?subject ?predicate ?person
} UNION { ?person rdf:type dbpedia-owl:Person .
?person ?predicate ?object }}
```

Transducer outcome - DBpedia

```
SELECT DISTINCT ?predicate WHERE { {
?person rdf:type foaf:Person .
?subject ?predicate ?person
} UNION { ?person rdf:type foaf:Person .
?person ?predicate ?object }}
```

Query 48(SP2B - Q10) - SP2B

```
SELECT ?subject ?predicate WHERE { ?subject ?predicate person:Paul_Erdoes }
```

Gold Standard - DBpedia

```
SELECT ?subject ?predicate WHERE {
?subject ?predicate dbpedia:Paul_Erdoes }
```

Transducer outcome - DBpedia

```
SELECT ?subject ?predicate WHERE {
?subject ?predicate dbpedia:Paul_Erdoes }
```

Query 49 (Dbpedia log) - DBPedia

```
SELECT ?y WHERE { ?y dbpedia-owl:binomialAuthority dbres:Johan_Christian_Fabricius. } LIMIT 10
```

Gold Standard - LIBRIS

```
SELECT ?y WHERE {
?y ?p libris:335843
} LIMIT 10
```

Transducer outcome - LIBRIS

```
SELECT ?y WHERE {
?y ?p libris:335843
} LIMIT 10
```

Query 50(SP2B - Q1) - SP2B

```
SELECT ?yr WHERE { ?journal rdf:type sp:Journal . ?journal dc:title "Journal 1 (1940)"8sd:string .
?journal dcterms:issued ?yr }
```

Gold Standard - DBPedia

```
SELECT ?yr WHERE {
?journal rdf:type dbpedia:PeriodicalLiterature.
?journal dc:title "Journal 1 (1940)"8sd:string .
?journal dcterms:issued ?yr }
```

Transducer outcome - DBPedia

```
SELECT ?yr WHERE {
?journal rdf:type dbpedia:AcademicJournal .
?journal dc:title "Journal 1 (1940)"8sd:string . }
?journal dcterms:issued ?yr }
```

A.8 Processing times for the query set

TABLE A.17: Segmented query processing times in *ms*.

Query	TT	AT	TAT	GSAT	AT–GSAT	E	H	A	P	F	Rule[Num]
Q1	8926	214	9140	227	-13	921			8005		E[1] P[1]
Q2	8525	652	9177	621	31	746			3967	3812	E[1] P[2] F[1]
Q3	2991	108	3099	123	-15				2991		P[1]
Q4	2005	364	2369	349	15	785	1220				E[1] H[1]
Q5	3500	405	3905	337	68			3500			A[1]
Q6	2688	210	2898	327	-117			2688			A[1]
Q7	4261	573	4834	641	-68	724			4110		E[2] P[2]
Q8	1142	847	1989	823	24	904			1085		E[1] P[1]
Q9	981	1021	2002	1038	-17	823			1179		E[1] P[1]
Q10	1802	938	2740	837	101	808	912			1020	E[1] H[1] F[1]
Q11	0	734	734	563	171						-
Q12	2825	208	3033	271	-63	995			1642		E[2] P[1]
Q13	167	290	457	304	-14	457					E[1]
Q14	268	312	580	289	23	479					E[1]
Q15	137	205	342	201	4	338					E[1]
Q16	330	277	607	327	-50	607					E[1]
Q17	1144	1482	2626	1105	377	342	1080			1204	E[1] H[1] F[1]
Q18	11264	523	11787	438	85				5253	6011	P[1] F[1]
Q19	1953	109	2062	306	-197	1953					E[1]
Q20	536	108	644	102	6	505					E[1]
Q21	1994	574	2568	388	186	1994					E[2]
Q22	5191	613	5804	592	21	1074			4117		E[1] P[2]
Q23	2753	409	3162	397	12	772			1981		E[1] P[1]
Q24	3503	102	3605	194	-92	1102				2401	E[1] F[2]
Q25	5658	689	6347	630	59	1767				3891	E[2] F[2]
Q26	3258	314	3572	309	5	805				2453	E[1] F[1]
Q27	1796	125	1921	128	-3	892				904	E[1] F[2]
Q28	2225	852	3077	821	31	511	827			887	E[1] H[1] F[1]
Q29	1153	901	2054	878	23					1153	F[1]
Q30	1765	782	2547	820	-38	684				1081	E[1] F[3]
Q31	1181	213	1394	308	-95	1181					E[1]
Q32	5899	1297	7196	1002	295	1991	893		3015		E[1] H[1] P[1]
Q33	6749	708	7457	1104	-396	855			5894		E[1] P[2]
Q34	5096	514	5610	665	-151	599			4497		E[1] P[2]
Q35	4658	160	4818	238	-78					4658	F[1]
Q36	4005	898	4903	759	139	817			3188		E[1] P[1]
Q37	3013	1650	4663	1702	-52					3013	E[2] P[1]
Q38	571	294	865	236	58	571					E[1]
Q39	4512	1834	6346	1463	371	1208		3304			E[1] A[1]
Q40	2899	775	3674	701	74				2325	574	P[1] F[1]
Q41	1103	1028	2131	991	37					1103	F[1]
Q42	4020	383	4403	293	90			2784	1236		A[1] P[1]
Q43	1029	469	1498	394	75					1029	F[2]
Q44	3598	493	4091	526	-33	527				3071	E[1] F[2]
Q45	0	197	197	176	21						-
Q46	1779	1328	3107	1128	200	571				1208	E[2] F[1]
Q47	0	122	122	255	-133						-
Q48	461	203	664	197	6	461					E[1]
Q49	3613	2751	6364	2809	-58	396				3217	E[1] F[1]
Q50	653	272	925	207	65		653				H[1]

Appendix B

Estimating query rewriting quality over LOD

B.1 Introduction

The increasing adoption of the Linked Open Data (LOD) paradigm has generated a distributed space of globally interlinked data, usually known as the Web of Data. This new space opens up the possibility of querying over a huge set of updated data. However, many users find difficulties when formulating queries over it, due to the fact that they are not familiar with the data, links and vocabularies of many heterogeneous datasets that constitute the Web of Data. In this scenario it becomes necessary to provide the users with tools and mechanisms that help them to exploit the vast amount of available data.

We can find in the specialized literature different proposals that have considered the goal of facilitating the task of querying heterogeneous datasets. We can highlight three main approaches among those proposals: 1) those that generate a kind of *centralized repository* that contains all the data of different datasets and then queries are formulated over that repository (e.g. [OD15]); 2) those that follow the *federated query* processing approach (e.g. [HMZ10]) in which a query against a federation of datasets is split into sub-queries that can be answered in the individual nodes where datasets are stored; and 3) those that follow the *exploratory query* processing approach (e.g. [HBF09]), which take advantage of the dereferenceable IRIs principle promoted by Linked Data. In this approach, query execution begins in a source dataset and is intertwined with the traversal of the HTTP dereferenceable IRIs to retrieve more data, from different nodes, that incorporate additional data for answering parts of the query and include more IRIs that can be successively dereferended to augment the queried dataset until the initial query is sufficiently answered. The two first approaches require a costly preparation task and the last one is mainly oriented to leverage the dereferencing architecture of Linked Data.

In the centralized and federated approaches, users pose queries using the vocabulary chosen for the global schema and can only expect answers from the centralized repository or from federated datasets. However, it is very common that several datasets offer data on the same or overlapped

domains. For example, GeoData and Geo Linked Data in the geographic domain, BNE (Biblioteca Nacional de España) and BNF (Bibliothèque National de France) in the bibliographic domain, MusicBrainz and Jamendo in the music domain, or Drugbank and Diseasome in the bio domain. Each centralized repository or a datasets federation only considers a limited collection of datasets and, therefore, cannot help a user with datasets that are out of the collection. Moreover, it seems interesting for the users to pose queries to a preferred dataset whose schema and vocabulary is sufficiently known by them and then a system could help those users enriching the answers to the query with data taken from a domain sharing dataset although with different vocabulary and schema. Our approach considers that type of systems. Notice that a proper rewriting of the query must be eventually managed by those systems.

In general, those kind of systems can be very useful in different scenarios. For example, an ordinary user posing a keyword-based query to a question answering system, which constructs a SPARQL query to be run on a source dataset, and then demanding for more answers from a dataset with different vocabulary. Another scenario can be that of scientists formulating queries over source datasets they are familiar with, and then, demanding more answers by accessing other different datasets, not requiring strict query equivalence but giving a chance to serendipity (notice that scientists need not be aware of the internal structure/vocabulary of the new target datasets). A third scenario can be that of an application programmer trying to query the English DBpedia using terms extracted from the user defined Spanish Wikipedia infoboxes (or whatever language Wikipedia), which are not mapped with official DBpedia terms. Then, in order to get some answers, a transformation of the source query is needed in order to be adequately expressed for the English DBpedia. The relevant common feature of all these scenarios is the need to cope with the vocabulary and schema heterogeneity of the stored data. Notice that such heterogeneity may reach the conceptual level leading to different granularity knowledge and to the point that some notions are conceptualized in one dataset but not in the others.

Our system deals with a query rewriting process where the preservation of the semantics is not a strong requirement and therefore it considers semantics-preserving and non-semantics-preserving rewritings in order to increase the opportunities of getting results. When a non-semantics-preserving scenario is considered, the definition of a quality estimation of the rewritten query becomes crucial because the user needs to be aware of the confidence that can be deposited on the results obtained from the new dataset.

As a motivating example, let us imagine a user that is only familiar with the LinkedMDB vocabulary (a dataset about movies and their related people). This user asks for the films and the names of art directors working on those films directed by Woody Allen and performed by Sean Penn. The SPARQL query constructed by the user could be the following one:

```
PREFIX mdb:<http://data.linkedmdb.org/resource/movie/>
SELECT  DISTINCT ?movie ?name
```

```
WHERE {
  ?woody  mdb:director_name "Woody Allen".
  ?movie  mdb:director ?woody;
          mdb:actor ?actor;
          mdb:film_art_director ?art.
  ?actor  mdb:actor_name "Sean Penn".
  ?art mdb:film_art_director_name ?name. }
```

Listing B.1: Films and names of art directors working on those films directed by Woody Allen and performed by Sean Penn.

and the obtained results are listed on table B.1:

Table B.1: Query results from LinkedMDB.

?movie	?name
db:film/38778	"Tom Warren"

Given the scarcity of the response or its inadequacy, the user would find useful to execute the same query in other datasets, perhaps more recognized ones or more active ones, trying to obtain more results. A good example of those datasets may be DBpedia. Using our system the user could obtain the following reformulation of the query, according to the DBpedia vocabulary:

```
PREFIX dbo:<http://dbpedia.org/ontology/>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT  DISTINCT ?movie ?name
WHERE {
  ?woody  foaf:name "Woody Allen"@en.
  ?movie  dbo:director ?woody;
          dbo:starring ?actor.
          dbo:cinematography ?art.
  ?actor  foaf:name "Sean Penn"@en.
  ?art foaf:name ?name. }
```

Listing B.2: Films and names of cinematographers working on those films directed by Woody Allen and performed by Sean Penn.

This reformulation was based on some declared mapppings. In particular, mdb:director was declared an equivalent property to dbo:director in a set of RDF triples published as an Open Linked Data file[1], captured from the web, and incorporated into a local Open Link Virtuoso RDF Store which allows to access them through a local SPARQL endpoint. Moreover, properties mdb:director_name, mdb:actor_name and mdb:film_art_director_name were declared as subproperties of foaf:name in the form of mappings by the own LinkedMDB dataset[2]. Although no declared mapping for mdb:film_art_director was found, the system proposed the term dbo:cinematography as an approximation to the original one, the same happened with the properties: mdb:actor and dbo:starring (in section B.4 we will show how this kind of proposed approximations can be discovered). The results obtained by the query in listing B.2 are presented on table B.2:

[1] http://wifo5-03.informatik.uni-mannheim.de/bizer/r2r/examples/ mappings.ttl
[2] http://wiki.linkedmdb.org/Main/Interlinking

TABLE B.2: Query results from DBpedia.

?movie	?name
dbr:Sweet_and_Lowdown	"Zhao Fei"
dbr:Sweet_and_Lowdown	赵非

Notice that new results appeared when querying the DBpedia dataset, which may be of interest to the user. Moreover, which further enriches the answer is the provision of some quality estimation of the reformulated query. It is at this point where a main contribution of this paper plays a relevant role. In this particular case, our system offered a *similarity factor* of 0.86 and a *quality score* of 0,79. The meaning of these features will be explained in subsequent sections.

In general, quality estimation can be defined in terms of a query similarity measure between the source and target queries (source query, formulated over the initial dataset, and target query, formulated over another dataset of the Web of Data indicated as target). However, when comparing two queries, different similarity dimensions can be considered [DG13]: (1) the query structure, expressed as a string or a graph structure; (2) the query content, its triple patterns and ontological terms and literal values; (3) the language features, such as query operators and modifiers; and (4) the results set retrieved by the query. Queries may be assessed with respect to one or several of that considered dimensions. And it is widely accepted that the application context heavily determines the choice for a similarity measure. We think that the more relevant dimensions to be taken into account in the scenario considered in this paper are (2) the content and (4) the result set. After all, structure and language features of a target query seem to be of less relevance to inform the user about the intended similarity spirit to the formulated source query. Therefore, (1) structure and (3) language dimensions are not considered in the proposed assessment. Although the result set is what matters to the user, it is crucial to notice that the intention of issuing the query to a target dataset is to look for more or different results than those obtained by the source query. Therefore, the intended result set of the target query cannot be compared with that of the source query in terms of exact matching and the query similarity measure must take into account this distinctive feature.

In summary, a main contribution of this paper is an approach that estimates the quality of rewritten queries, including non-semantics-preserving rewriting, to search the distributed space of globally interlinked data, that is different from the *centralized repository*, *federated*, and *exploratory* aforementioned approaches. Terms for the source query can be selected from a preferred schema or vocabulary instead of that offered by the centralized approach or federated schema. Moreover, target datasets are not limited to those comprising the centralized repository or the federation. Technical contributions to support the proposed approach are: (1) A proposal of a general framework for the deployment and management of a query rewriting system concerning the scenario previously explained. And, (2) the validation of the framework in a real context, including the machine

learning and optimization techniques used to estimate the quality of the rewriting outcome.

Proposal of a general Framework. We propose a new framework that groups the following components: query rewriting rules, an algorithm to manage the rules, a set of similarity measures, and a predictive model. This framework would be oriented to two types of users:

- *End users*, who formulate a query over a source dataset and then the system issues a new query, which mimics the original one, over a target dataset (of the Web of Data) whose results further enrich the answer. The new issued query is annotated with a similarity factor between queries and a quality score.

- *Expert/technical users* who, in addition to benefiting from the functionalities provided for end users, can also include in the framework: new rewriting rules, algorithms to process them, and similarity measures to qualify the query rewriting. The framework would also provide them with facilities to tune the introduced similarity measures by means of optimization techniques. The rewriting rules, similarity measures and training queries introduced could be stored in the log of the framework with the idea of serving as experimental and comparison benchmark.

Validation of the Framework. The framework has been tested in a real context. For that, we have instantiated the framework with the following elements:

- Query rewriting rules. Apart from some rules dealing with the rewriting of terms by their specified equivalents, via synonym mappings or EDOAL (Expressive and Declarative Ontology Alignment Language) [ESZ07] alignment rules, the framework also deals with some other heuristic based rules which conform a carefully controlled set of cases.

- An algorithm to schedule the application of the rules.

- Similarity measures. The computation of the similarity factor takes into account different similarity measures depending on the motif of the rule being applied. Those motifs range from relational to ontological structure, and from language based to context based similarity.

- Queries. 100 queries were formulated over the previously selected datasets. Three domain areas were considered for the datasets: media-domain, bibliographic, and life science. From the media-domain six datasets were selected, five datasets from the bibliographic domain, and five more from the life science domain, respectively. When selecting the queries, our aim was to get a set that would contain a broad spectrum of SPARQL query types [AFMPdlF11]. Concerning provenance we selected queries that appeared in well known benchmarks such as QALD4 or FedBench, and we also considered queries that belonged to LOD SPARQL endpoints logs from the selected datasets.

- Predictive model. A model has been created using a supervised machine learning method applied to the considered experimental scenario to predict the F1 score of each rewritten query.

The rest of the paper is organized as follows. Section B.2 presents some related work in the scope of resource matching and query rewriting. Section B.3 introduces a description of the main features of the proposed framework. Section B.4 shows a framework embodiment. Section B.5 describes the framework validation results. Finally, some conclusions are presented.

B.2 Related work

The impressive growth of the Web of Data has pushed the research on Data Linking [SFN11]: "the task of determining whether two object descriptions can be linked one to the other to represent the fact that they refer to the same real-world object in a given domain or the fact that some kind of relation holds between them". Those object descriptions can be expressed with diverse structural relationships, depending on different contexts, using classes and properties from different ontologies. Research on object similarity and class matching has issued a considerable amount of techniques and systems in the field of Ontology Matching [ES13b], although less work has been devised for property alignment [CH14, LTLL09]. The work in [KM15] presents an unsupervised learning process for instance matching between entities. Queries considered in this paper involve terms for classes, properties and individuals. Therefore, techniques for discovering similarity for any of them are relevant. However, the topic of this paper regards query similarity, which can be recognized as a different problem. As has been noticed in [DG13], the appropriate notion of query similarity depends on the goal of the task. In our case, the task is to estimate the similarity of the intended semantics between a query designed for a source dataset and a rewriting to a different vocabulary, to be evaluated in a different target dataset.

Some works, for example [CSM+10a, MBGC12a], have approached a restricted version of the task carried out in our case. They restrict themselves to produce semantic preserving translations (i.e. total similarity) and so they assume that enough equivalent correspondences exist among entities in datasets. Taking into account that such an assumption is too strong in real scenarios we consider situations where different types of correspondences exist (not only of equivalence type) and even more, situations where some correspondences are missing. This consideration implies that query semantics is sometimes not preserved in the rewriting process and therefore the estimation of similarity of the produced rewriting becomes crucial.

The aim of our considered rewriting is to look for more answers in a target dataset than those obtained from the source dataset. Some other works have the goal of obtaining more answers (including approximate ones) for an original query; however, all of them restrict their scope to

a single source dataset. In [HPW08] they propose a logical relaxation of conditions in conjunctive queries based on RDFS semantics. Those conditions are successively turned more general and a ranking in the successively obtained answers is generated. [HLZ08, HLZ12] use the same kind of relaxations as [HPW08], but propose different ranking models. In [HLZ08], similarity of relaxed queries is measured with a model based on the distance between nodes in the ontology hierarchy. In [HLZ12], they use an information content based model to measure similarity of relaxed queries. The work in [ERW11] addresses the query relaxation problem by broadening or reformulating triple patterns of the queries. Their framework admits replacement of terms by other terms or by variables and also removal of entire triple patterns. In that work, generation and ranking of relaxed queries is guided by statistical techniques: a distance between the language models associated to entity documents is defined. All those works can be situated under the topic of query relaxation.

With different use cases in mind, the papers [DVMT13, HMPS12, PSW16] present different possibilities for approaching the querying of Linked Data. In [HMPS12] a framework for relaxation of star-shaped SPARQL queries is proposed. They present different *matchers* (functions that map pairs of values to a relaxation score) for different kinds of attributes (numeric, lexical or categorical). The framework may involve multiple matchers. The matchers generate a tuple of numeric distances between a query and an entity (answer for the query). Notice that the distance is defined between an entity and a query, not between two queries as in our approach. [DVMT13] proposes a measure to evaluate the similarity between a graph representing a query and a graph representing the dataset. With a suitable relaxation of the notion of alignment between query graph paths and dataset graph paths they generate approximate answers to queries. In [PSW16] a method for query approximation, query relaxation, and their combination is proposed for providing flexible querying capabilities that assist users in formulating queries. Query answers are ranked in order of increasing distance from the user's original query.

In summary, cited works that transform the query or reformulate the notion of answer in order to provide users with more answers from the source dataset, do not try to reformulate the query in a different dataset with different vocabulary and data structure; and this is a distinguishing feature of our use case.

It is worth mentioning another data access paradigm that uses query rewriting. In the Ontology Based Data Access (OBDA) paradigm, an ontology provides a conceptual view of the data and a vocabulary for user queries [PLC+08]. Users pose queries in terms of a convenient conceptualization and familiar vocabulary, without being aware of the details of the structure of data sources. SPARQL can be considered as a query language in this paradigm [CCK+17], and the SPARQL query must be rewritten in an appropriate query language for the underlying data source which, for instance, could be SQL for relational databases. Such rewritting is based on mappings between terms in the ontology and (in case of relational

databases) views of the relational schema. R2RML [C+12] is a W3C standard language for specifying those mappings. A sufficiently complete set of mappings must be specified in order to rewrite the query, since OBDA paradigm intends to process a query, over the underlying data source, which is semantically equivalent to the query posed by the user with the ontology vocabulary. Notwithstanding the relevance of OBDA paradigm, we point out that it tackles with a different problem to the stated one in this paper. OBDA rewrites a query to adapt it to another data model. The problem tackled in this paper is to rewrite a SPARQL query to adapt it to another vocabulary without considering complete mappings between the respective vocabularies.

B.3 Abstract framework

An abstract representation of the proposed framework for rewriting a query and estimating the quality of the rewritten query, can be expressed as a structure ($\mathcal{R}$, $\mathcal{A}$, $\mathcal{Q}$, $\mathcal{P}$) where

- $\mathcal{R}$ is a set of SPARQL query rewriting rules,
- $\mathcal{A}$ is the algorithm for applying the rules,
- $\mathcal{Q}$ is a rewriting quality estimation system, composed of three elements ($\mathcal{M}$, $\mathcal{V}$, $\mathcal{SF}$) such that
  - $\mathcal{M}$ is a set of similarity measures between fragments of query expressions,
  - $\mathcal{V} : \mathcal{R} \rightarrow \mathcal{M}$ is an application that associates a similarity measure to each rule, and
  - $\mathcal{SF} : \mathcal{R}^* \rightarrow [0, 1]$ associates each sequence of applied rewriting rules with a similarity factor from the $[0, 1]$ real interval,
- $\mathcal{P}$ is a predictive model which estimates a quality score for the target query.

The part of a SPARQL query to be rewritten by rules in $\mathcal{R}$ is the graph pattern in the WHERE clause of the query. A graph pattern consists of a set of *triple patterns*. A *triple pattern* is a triple (s, p, o) where s is the subject, p is the predicate, and o is the object. The three of them represent resources and any of them can be a *variable* (denoted by prefixing it with a question mark, for instance $?x$). The rule language is a variation of the CONSTRUCT query form of SPARQL 1.1, as follows:

```
REPLACE  template
BY       template
WHEN {
    graph pattern
}
```

The REPLACE clause presents a *template* that should be matched to a part of the graph pattern in the query being rewritten. This matching

is the trigger of the rule. A *template* is a graph pattern including three kinds of *tokens*: *IRI tokens*, *variable tokens*, and *wild tokens*. A *IRI token* only binds to IRIs in the graph pattern of the query, a *variable token* only binds to variables, and a *wild token* binds to both. IRI tokens are prefixed by *s:* or *t:* meaning that they only bind to IRIs in the source or target dataset, respectively. Variable tokens are prefixed with a question mark. Wild tokens are prefixed with a hash, for instance *#u*. The matched part in the graph pattern of the query will be replaced by the binded template in the BY clause if the graph pattern in the WHEN clause find matches with the data graph of the datasets in question (the BY clause resembles the CONSTRUCT clause and the WHEN clause resembles the WHERE clause in SPARQL queries, but for replacement of triple patterns in a graph pattern).

For instance, the following rule:

```
PREFIX s:<source dataset>
PREFIX t:<target dataset>
REPLACE  #s s:p  #o .
BY       #s t:p  #o .
WHEN {
   s:p owl:sameAs t:p .
}
```

applied to the query in listing B.1, produces the query

```
PREFIX mdb:<http://data.linkedmdb.org/resource/movie/>
PREFIX dbo:<http://dbpedia.org/ontology/>
SELECT  DISTINCT ?movie ?name
WHERE {
  ?woody   mdb:director_name "Woody Allen".
  ?movie   dbo:director ?woody;
           mdb:actor ?actor;
           mdb:film_art_director ?art.
  ?actor   mdb:actor_name "Sean Penn".
  ?art   mdb:film_art_director_name ?name. }
```

LISTING B.3: mdb:director replaced with dbo:director.

by rewriting the triple pattern (?movie mdb:director ?woody) by (?movie dbo:director ?woody) due to the appearance of (mdb:director owl:sameAs dbo:director) in the consulted data graph, in this particular case in the local Virtuoso RDF store.

This rule language is sufficiently expressive since WHEN clauses can use the full expressivity of graph patterns in SPARQL 1.1. The core of the implementation of those rules can be supported by an almost direct generation of SPARQL queries from the rule expression. For instance, the query supporting the previous sample rule could be constructed in the following way: Assume that the template *(#s s:p #o)* (appearing in the REPLACE clause) matches a part of the graph pattern in the query being rewritten and yields a binding of the IRI token *s:p* to the IRI *s:IRIp* in the source dataset. Then, the triple pattern (*s:IRIp owl:sameAs ?t0p*) (originated from the WHEN clause) will be the WHERE clause and *?t0p* will be the projected variable in the SELECT clause due to the BY clause in the rule (see Listing 4).

```
SELECT  ?t0p
WHERE {
 s:IRIp owl:sameAs ?t0p .
}
```

Listing B.4: Query supporting rule implementation.

Then, the results of that query can be used to form the corresponding replacements specified in the BY clause.

Rules in this paper only consider the basic RDF entailment regime. Nevertheless, if the mediating SPARQL endpoint managing the query processing implements another entailment regime over the dataset of interest, the rewriting process leverages on that enriched regime without any harm.

A rewriting rule set requires an algorithm to manage the rewriting process, this is the role of element $\mathcal{A}$ in the abstract framework. Different algorithms managing the same set of rules may produce different outcomes.

The rewriting system of the proposed framework takes a given query Q_s (named *source query*), expressed with a vocabulary adequate[3] for the source dataset, and transforms it into another query Q_t (named *target query*), expressed with a vocabulary adequate for the selected target dataset. In this paper, the primary problem of a single target dataset is considered, although the process may be iterated with a different target dataset each time. Decomposition of the source query into parts and distribution of each part to a different target dataset is devoted to future work. However, it should be noted that it could be solved by combination of solutions of the primary problem. The rewriting process produces Q_t as a semantically equivalent query to Q_s as long as enough equivalence mappings between the vocabulary of the source dataset and the vocabulary of the target dataset are found. But the distinguishing point is that the process produces a mimetic query Q_t even in the case when no equivalent translation for Q_s is found. That is to say, semantic preservation cannot be guaranteed due to vocabularies heterogeneity and missing links with terms appearing in the source query. It is at this point where the definition of a quality estimation of the rewriting outcome becomes crucial to our approach, because the user needs to be aware of the quality of the produced target query. This is the goal of the Q element in the abstract framework.

Every application of a rule r is considered as a step in the progress to the target query, and such steps are valuated with a factor computed by the associated similarity measure $\mathcal{V}(r)$ from $\mathcal{M}$.

The function $\mathcal{SF}$ calculates a similarity factor for a target query in terms of the sequence of rules $\bar{r}$ that were applied to construct it and properly combining the measures $\mathcal{V}(r)$ (for each $r \in \bar{r}$). Similarity measures in $\mathcal{M}$ can be defined by simple functions or very complex ones. Usually they can be defined by combining similarity measures taken from a state-of-the-art repository [ES13b].

As previously said in the introduction section, the intention of issuing a query to the selected target dataset is to look for more or different results than those obtained in the source dataset. Therefore, although the target

[3] We say that a term is *adequate* for a dataset if its IRI prefix follows the proprietary format of the dataset or it appears in the dataset vocabulary.

query should try to maintain the spirit of the source query, the intended result set of the target query cannot be compared with the source query retrieved set but with that of an *ideal* expression of such source query in terms of the vocabulary acceptable by the target dataset. Notice that, due to the previously mentioned heterogeneity reasons, such ideal expression cannot be trivially constructed. In fact, we consider that the finding of such ideal expression, in the considered scenario, should be realized by a human expert who knows vocabularies of source and target datasets. And, therefore, the reference query against which the target query should be compared is a human designed one, that tries to express the most similar intention to the source query but in the context of the target dataset. We consider such a query our *gold standard* query against which the target query should be compared.

In the presence of a gold standard query, its results can be compared with those obtained by the target query. Statistical measures such as precision, recall and F1 score can be used to measure the quality of a target query. Of course, gold standards can only exist in an experimental scenario but not in the real setting, and that is the reason to incorporate machine learning techniques in the framework. The predictive model $\mathcal{P}$ is generated by a supervised machine learning method applied to a suitable experimental scenario consisting of a selected benchmark of source queries with their respective gold standard queries for the target datasets, and the set of corresponding target queries generated by the rewriting system with their respective $\mathcal{SF}$ value and with their respective F1 score that will be the goal for prediction.

Note that this framework establishes, principally, a scenario for experimentation, where different materializations of each element of the framework can be assessed and compared.

In particular, it should be taken into account that the provision of gold standard queries involves a delicate work: knowledge of different vocabularies is needed and sometimes different choices can be considered as appropriate gold standard of a query. Furthermore, the production of desired quantities of training queries is a time consuming task. The editors of the gold standard queries considered in the experiment reported in this paper were just the authors of the paper. Queries were grouped by domain and each domain group was assigned to a different author. Then, the work was reviewed jointly. Therefore, the results of our experiments could be considered to be improved if we had a larger and much more supervised collection of queries.

B.4 Framework embodiment

This section presents a brief explanation of a specific embodiment of the abstract framework ($\mathcal{R}$, $\mathcal{A}$, $\mathcal{Q}$, $\mathcal{P}$) that was partially presented in [TBBI15] and which serves as a proof of concept for our proposal.

The set of rules $\mathcal{R}$ was devised from a pragmatic point of view. The rules set up common sense heuristics to obtain acceptable rewritings even when no semantically equivalent translations are at hand. Preconditions

for the application of the rules take into account a carefully restricted context of the terms occurring in the graph pattern. Although restricted, the rule set has shown to be quite effective achieving acceptable rewritings (see section B.5).

Five kinds of rules have been considered, each kind based on a different motif: Equivalence (E), Hierarchy (H), Answer-based (A), Profile-based (P), and Feature-based (F).

Furthermore, a pragmatic scenario has been considered in which a bridge dataset can be taken into account in the process of rewriting a query adequate for a source dataset into another query adequate for a target dataset. In order to favour the possibilities of finding alignments between resources, mappings between both the source (D_s) and target (D_t) datasets and a bridge (D_b) dataset are considered. Such a choice is justified because that scenario is quite frequent, since in almost any domain there is a popular dataset that may play such a reference role. For instance: BabelNet in the linguistic domain, DBLP in the Computer Science Bibliographic domain, NCI Thesaurus in the clinical domain, New York Times-Linked Open Data in the media domain, reference.data.gov.uk in government domain, or Dbpedia in cross domain. There is not a fixed bridge dataset for each domain, any dataset may play the role of bridge dataset for each occasion instead. More ambitious scenarios may consider bridge concatenations, but a balance between computational cost and completeness decided us for restricting to only one bridge dataset per rule application.

Equivalence rules basically consist in replacing a query fragment by an equivalent one. They are the most frequent kind of query rewriting rules in the technical literature. Of course, their use is the most reasonable decision when such equivalence mappings are at hand; and can be confident that such rewriting preserves the semantics of the query. In section B.3 a simple equivalence rule regarding the predicate of a triple pattern was applied to the source query in listing B.1. Next, the expression of another of our equivalence rules is presented, namely one that replaces the subject of a triple pattern. Notice that a bridge dataset is used and diverse equivalence mappings are considered (see the FILTER clauses):

```
PREFIX s:<source dataset>
PREFIX t:<target dataset>
PREFIX b:<bridge dataset>
REPLACE   s:u  #p  #o .
BY        UNION(t:u #p  #o)
WHEN {
    s:u  ?eq1   b:u .
    b:u ?eq2   t:u  .
    FILTER (?eq1 = owl:sameAs ||
              ?eq1 = owl:equivalentClass ||
              ?eq1 = owl:equivalentProperty ||
              ?eq1 = skos:exactMatch)
    FILTER (?eq2 = owl:sameAs ||
              ?eq2 = owl:equivalentClass ||
              ?eq2 = owl:equivalentProperty ||
              ?eq2 = skos:exactMatch)
}
```

where UNION(t:u #p #o) represents the UNION pattern of all the triple patterns constructed with the IRIs binded with t:u.

The similarity measure associated to equivalence rules (E) is simply the constant function $\phi(u) = 1$, representing the semantics preservation after the replacement of the non adequate term u.

Hierarchy rules consist in replacing a term by a semantic generalization or restriction of that term. Such kind of rules are considered in works that account for relaxing or narrowing queries. In cases where equivalence is not guaranteed, replacing a term by its most specific subsumer or its most general subsumee expression changes the semantics in a ontological biased way. Next, the expression of one of our hierarchy rules is presented:

```
PREFIX s:<source dataset>
PREFIX t:<target dataset>
REPLACE  #s  s:p  #o .
BY       AND(#s t:p  #o)
WHEN {
   s:p  ?sub  t:p .
   FILTER (?sub = rdfs:subPropertyOf ||
           ?sub = skos:narrower)
}
```

where AND(#s t:p #o) represents the conjunction pattern of all the triple patterns constructed with the IRIs binded with t:p. Three successive applications of this hierarchy rule to the query in listing B.3 rewrites it to the following query:

```
PREFIX mdb:<http://data.linkedmdb.org/resource/movie/>
PREFIX dbo:<http://dbpedia.org/ontology/>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT  DISTINCT ?movie ?name
WHERE {
  ?woody  foaf:name "Woody Allen".
  ?movie  dbo:director ?woody;
          mdb:actor ?actor;
          mdb:film_art_director ?art.
  ?actor  foaf:name "Sean Penn".
  ?art   foaf:name ?name.  }
```

LISTING B.5: mdb:director_name, mdb:actor_name and mdb:film_art_director_name replaced with foaf:name.

by rewriting the properties mdb:director_name, mdb:actor_name and mdb:film_art_director_name by the property foaf:name due to the appearance of (mdb:director_name rdfs:subPropertyOf foaf:name), (mdb:actor_name rdfs:subPropertyOf foaf:name) and (mdb:film_art_director_name rdfs:subPropertyOf foaf:name) as mappings provided by the own LinkedMDB dataset[4].

Similarity estimation of hierarchy related terms is usually based on a distance measure. It is generally considered that the depth associated to the compared terms in the hierarchy influences the conceptual distance between the terms. Low depth correspond to more general terms and

[4] http://wiki.linkedmdb.org/Main/Interlinking

high depth correspond to more specific terms. Nearby high depth terms tend to be more semantic similar than low depth ones. Consequently, the similarity function selected for hierarchy rules (H) was an adaptation of a distance proposed in [ZZLY02] and elsewhere. Each term u in the hierarchy is associated with a *milestone* value *m(u)* depending on its depth in the hierarchy. Then, the distance between two terms u and v in the hierarchy is $d(u, ccp(u, v)) + d(v, ccp(u, v))$ where *ccp(u,v)* is the *closest common parent* of u and v in the hierarchy and $d(x, ccp(x, y)) = m(ccp(x, y)) - m(x)$. The milestone of a term u is defined as:

$$m(u) = \frac{1}{2 \times k^{depth(u)}}$$

where k is a predefined factor bigger than 1 that indicates the rate at which the value decreases along the hierarchy, and $depth(u)$ is the length of the longest path from the node u to the *root* ($depth(root) = 0$). In this case we used $k = 2$.

However, in the context considered in this paper, where u and v belong to different vocabularies and, therefore, different hierarchies, the notion of *ccp(u,v)* is not directly applicable. Then, we have adapted such distance. The intuition behind is that the deepest term (i.e. that with the least milestone) carries more information than the higher term:

$$\begin{aligned} distance(u, v) &= m(u) \times (1 - \frac{1}{k}) \\ &\quad \text{if } m(u) = m(v) \wedge u \sqsubseteq v \\ distance(u, v) &= m(v) - m(u) \\ &\quad \text{if } m(u) < m(v) \wedge u \sqsubseteq v \\ distance(u, v) &= m(u) \times (1 - \frac{1}{k}) \\ &\quad \text{if } m(u) < m(v) \wedge v \sqsubseteq u \end{aligned}$$

Once the distance is defined, the similarity function S_o between the terms u and v is

$$S_o(u, v) = 1 - distance(u, v)$$

The number of terms involved in the replacement of a term u and its triple pattern can be more than one, depending on the particular bindings of the applied rewriting rule. In such a case, if there are n bindings $(u_1, \ldots, u_n)$, the similarity measure is the average of the n pairwise similarity values:

$$\phi(u) = \frac{\sum_{i=1}^{n} S_o(u, u_i)}{n}$$

In the particular case of the current query example, it resulted $\phi(mdb : director\_name) = 0.6$, $\phi(mdb : actor\_name) = 0.6$ and $\phi(mdb : film\_art\_director\_name) = 0.6$.

Equivalence and hierarchy rules can be applied when direct mappings for the term to be replaced are found. But, if the involved datasets are not

completely aligned and those direct mappings are missing, new kinds of rules trying to leverage mappings of terms surrounding the focussed term should be considered. And that is precisely what our proposed Profile-based, Answer-based, and Feature-based kinds of rules do.

Let us call the *profile* of a resource x in a dataset D to the set of resources that are related to x, as subjects or objects, through triples in D. More specifically:

$$\begin{aligned}\mathcal{P}_D(x) \quad = \quad & \{v \in Terms(D) \mid \\ & (\exists p.(x,p,v) \in D \vee (v,p,x) \in D) \vee \\ & (\exists a.(a,x,v) \in D \vee (v,x,a) \in D)\}\end{aligned}$$

The heuristic considered in **Profile-based** rules is the following: if a resource v, in the profile of the focused resource u, is equivalent to a resource $t : v$ in the target dataset, and there is a resource $t : u$ in the profile of $t : v$, sufficiently similar to u, then u could be replaced by $t : u$.

For instance, the following triples:

```
mdb:film/38778  mdb:film_art_director        mdb:
    film_art_director/238 .
mdb:film/96785  mdb:film_art_director        mdb:
    film_art_director/1 .
mdb:film/38180  mdb:film_art_director        mdb:
    film_art_director/2 .
mdb:film_art_director/84  rdf:type           mdb:
    film_art_director .
mdb:film_art_director/363 rdf:type           mdb:
    film_art_director .
```

are only some of the triples in LinkedMDB that would determine the profile of mdb:film_art_director. Considering only such small set, the profile would be the set:

```
{mdb:film/38778,  mdb:film_art_director/238,
 mdb:film/96785,  mdb:film_art_director/1,
 mdb:film/38180,  mdb:film_art_director/2,
 mdb:film_art_director/84,
 mdb:film_art_director/363}
```

For the sake of the example, let us consider only one of those resources in the profile. For instance, mdb:film/38778. A mapping of equivalence was found betweeen mdb:film/38778 and the resource dbr:Sweet_and_Lowdown in DBpedia. And some triples were found in DBpedia involving dbr:Sweet_and_Lowdown. For instance:

```
dbr:Sweet_and_Lowdown  dbo:cinematography    dbr:Zhao_Fei .
dbr:Sweet_and_Lowdown  dbo:director          dbr:Woody_Allen .
dbr:Sweet_and_Lowdown  dbo:distributor       dbr:
    Sony_Pictures_Classics .
dbr:Sweet_and_Lowdown  dbo:editing           dbr:Alisa_Lepselter
    .
dbr:Sweet_and_Lowdown  dbo:gross          4197015.0 .
dbr:Sweet_and_Lowdown  dbo:producer          dbr:Jean_Doumanian
    .
dbr:Sweet_and_Lowdown  dbo:runtime           5700.000000^^(xsd:
    double) .
```

The same process should be done with all the resources of the profile. Next, calculating the similarity between mdb:film_art_ director and any of the predicates appearing in the preceding set of triples we found the following values:

```
S(mdb:film_art_director, dbo:cinematography)=0.72
S(mdb:film_art_director, dbo:director)=0.61
S(mdb:film_art_director, dbo:distributor)=0.56
S(mdb:film_art_director, dbo:editing)=0.54
S(mdb:film_art_director, dbo:gross)=0.31
S(mdb:film_art_director, dbo:producer)=0.20
S(mdb:film_art_director, dbo:runtime)=0.18
```

Then, the resource with the maximum similarity value was selected to replace mdb:film_art_director, namely dbo:cinematography. And the value of the similarity measure was $\phi(mdb : film\_art\_director) = 0.72$.

Next, the expression of the profile rule responsible of the previous replacement is presented:

```
REPLACE  #s s:p  #o .
BY       #s t:p  #o .
WHEN {
    ?s s:p ?o .
    ?s ?eq ?ts .
    ?o ?eq ?to .
    {?ts t:p ?to .}
    UNION
    {?to t:p ?ts .}
    FILTER (?eq = owl:sameAs ||
            ?eq = skos:exactMatch)
    FILTER (t:p = maxSim(s:p, h, profile(s:p))
}
```

Regarding the notion of sufficient similarity, we decided to establish a threshold h to be exceeded by the value of a similarity function between resources $S(u, v)$, in order to consider v sufficiently similar to u. When several resources exceed the threshold, the resource with maximum similarity value is selected for the replacement. Let us denote $maxSim(u, h, R)$ to a resource w in the set of resources R which is the most similar to u and whose similarity value is greater than the threshold value h:

$$\begin{aligned} maxSim(u, h, R) &= w \\ \text{such that} \quad & w \in R, \\ & S(u, w) \geq h, \quad \text{and} \\ & \forall v \in R.\ S(u, w) \geq S(u, v) \end{aligned}$$

In the current framework, the similarity function of two resources is defined as a linear combination of some other three similarity measures, which are selected to compare a context of the terms.

$$\begin{aligned} S(u, v) &= \alpha_n \cdot S_n(u, v) + \alpha_d \cdot S_d(u, v) + \alpha_o \cdot S_o(u, v) \\ & \alpha_n, \alpha_d, \alpha_o \geq 0 \quad \wedge \quad \alpha_n + \alpha_d + \alpha_o = 1 \end{aligned}$$

S_n and S_d are string based methods, and S_o is the similarity measure previously defined. S_n is a similarity measure computed as the average of Levenshtein and Jaccard distances, corresponding to the rdfs:label property value of the two compared terms. And S_d takes into account the definition contexts of the terms: for each compared term, u and v, a bag of words is constructed containing words from their rdfs:comment and rdfs:label string valued properties. S_d is defined as a cosine similarity of two vectors $V(u)$ and $V(v)$ constructed by the frequency of word appearance (i.e. Vector Space Model technique):

$$S_d(u, v) = \frac{V(u) \cdot V(v)}{\|V(u)\| \, \|V(v)\|}$$

Definition of $S(u, v)$ can be considered simple if compared to functions that involve more sophisticated linguistic techniques, or use some other Information Content techniques, or take into account much more information about the resources. But we think that the computational cost of those alternatives must be carefully considered, given the use case scenario presented in the introduction of this paper. In spite of its simplicity, results of our experiments are encouraging for researching along that line (see section B.5).

Then, the similarity measure associated to u after the application of a Profile-based rule is

$$\phi(u) = S(u, maxSim(u, h, profile(u)))$$

This kind of rule was used to replace mdb:film_art_director by dbo:cinematography, and also mdb:actor by dbo:starring, in the working example presented in the introduction, with a $\phi(mdb : film\_art\_director) = 0.72$ and $\phi(mdb : actor) = 0.89$. The result is the query presented in listing B.2 which is completely expressed with the target dataset vocabulary.

However, after the application of some query rewriting rules, some non adequate terms of the query could still be unreplaced. The proposal in this paper considers two more kinds of rules in order to cover some more interesting circumstances. **Answer-based** rules are supported by bindings obtained, during the source query processing over the source dataset, for the piece of graph pattern to be replaced. Those binded resources are considered as examples of what the query is looking for in the target dataset. Triples involving those resources in the target dataset are used to mimic the triple pattern to be replaced. The intuition behind is that triples stated about the answer samples in the target dataset probably resemble expected answers of the original query.

For instance, consider the query

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?a ?p ?q
WHERE {
  dbr:The_Other_Side_of_the_Wind ?p ?a .
  ?a ?q dbo:Agent.
```

```
}
```

LISTING B.6: Information about The Other Side of the Wind.

for DBpedia as source dataset and consider LinkedMDB as the target dataset. Then, dbr:The_Other_Side _of_the_Wind and dbo:Agent are not adequate for LinkedMDB.

Consider the following set of triples in the source dataset (DBpedia):

```
dbr:The_Other_Side_of_the_Wind dbo:starring dbr:John_Houston.
dbr:The_Other_Side_of_the_Wind dbo:starring dbr:
    Peter_Bogdanovich.
dbr:The_Other_Side_of_the_Wind dbo:director dbr:Orson_Wells.
dbr:The_Other_Side_of_the_Wind dbo:cinematography
    dbr:Gary_Graver.
dbr:The_Other_Side_of_the_Wind dbo:starring dbr:Susan_Strasberg
    .
```

Consider the five bindings of variable ?a, in the first triple pattern of the query, as answer samples to that triple pattern. And consider also that those answer samples are mapped as equivalent to corresponding resources in the target dataset (LinkedMDB):

```
dbr:John_Houston owl:sameAs mdb:actor/29769.
dbr:Peter_Bogdanovich owl:sameAs mdb:actor/29762.
dbr:Orson_Wells owl:sameAs mdb:producer/9736.
dbr:Gary_Graver owl:sameAs mdb:actor/9677.
dbr:Susan_Strasberg owl:sameAs mdb:actor/37472.
```

Then, triples about those answer samples in the target dataset could probably resemble expected answers of the original query. For each one of those answer samples in the target dataset (LinkedMDB), triples as the following are found in the dataset:

```
mdb:actor/29769  mdb:actor mdb:film/133;
                 mdb:actor mdb:film/1025;
                 mdb:actor mdb:film/46921;
                 mdb:actor mdb:film/23486.
mdb:actor/29762  mdb:actor mdb:film/38395;
                 mdb:actor mdb:film/46921.
mdb:producer/9736  mdb:actor mdb:film/38078;
                   mdb:actor mdb:film/46921;
                   mdb:actor mdb:film/66530.
mdb:actor/9677  mdb:actor mdb:film/89003;
                mdb:actor mdb:film/46921.
mdb:actor/37472  mdb:actor mdb:film/274;
                 mdb:actor mdb:film/46921.
```

As a crude approximation, the most frequent resource appearing in such a context could be considered to replace the non adequate term dbr:The_Other_Side_of_the _Wind in the query. In this case, mdb:film/46921 appeared 11 times in the running experiment and was selected for the replacement. Then, its similarity value was calculated, yielding $\phi(dbr : The\_Other\_Side\_of\_the\_Wind) = 0.71$.

Then, the rewritten query obtained after applying that answer-based rule would be the following :

```
PREFIX mdb: <http://data.linkedmdb.org/resource/movie/>
SELECT DISTINCT ?a ?p ?q
WHERE{
    mdb:film/46921 ?p ?a .
    ?a ?q dbo:Agent .
}
```

Listing B.7: Information about movie:46921.

The rule expression capturing the process described above could be as follows:

```
PREFIX s:<source dataset>
PREFIX t:<target dataset>
REPLACE s:u ?p ?x
BY      t:u ?p ?x
WHEN {
    ?s as t:u (COUNT(?s) as ?OCCURNUM)
    WHERE {
        s:u ?p ?x .
        ?x ?eq t:?o .
        ?s t:?pt t:?o .
        FILTER (?eq = owl:sameAs ||
                ?eq = skos:exactMatch)
    }
    GROUP BY ?s ORDER BY DESC ?OCCURNUM
    LIMIT 1
}
```

Only a restricted set of templates is selected for applying the Answer-based rewriting rules. In particular, triple patterns where only one term remains non adequate for the target dataset. The other two terms of the triple pattern are an adequate term for the target dataset and a variable or else two variables. Namely, the selected templates are: (?x t:p s:u), (s:u t:p ?x), (?x s:p t:o), (t:o s:p ?x), (?x ?p s:o), (s:u ?p ?x). The adequate term may be there from the beginning (i.e. some terms can be adequate for both source and target dataset) or else be the result of a previously applied rewriting rule.

As has been previously noted, when the number of terms involved in the replacement of the term u is more than one (let us say $(t\colon b_1, \ldots, t\colon b_k)$), every single measure is replaced by the corresponding average $(S_x(u, t\colon b_1) + \ldots + S_x(u, t\colon b_k))/k$. Then, a linear combination of the same aforementioned similarity measures is associated to the replaced term u:

$$\begin{aligned}\phi(u) \quad = \quad & \alpha_n \cdot \frac{\sum_{i=1}^{k} S_n(u, t\colon b_1)}{k} + \\ & \alpha_d \cdot \frac{\sum_{i=1}^{k} S_d(u, t\colon b_1)}{k} + \\ & \alpha_o \cdot \frac{\sum_{i=1}^{k} S_o(u, t\colon b_1)}{k}\end{aligned}$$

Values for parameter α_n, α_d, and α_o could be determined by an expert taking into account the desired weighting of the three facets. But it could be preferable to obtain those parameter values as the output of a

specifically designed optimization algorithm. The explanation of the configuration of the optimization algorithm will be presented in section B.5, within the experimental scenario.

The last kind of rules we are considering in the present embodiment is **Feature-based** rules. This kind of rules is the last option if non adequate terms remain in the query graph pattern after the aforementioned kind of rules have already been considered. Notice that running a query with a non adequate term for the considered dataset would yield the empty answer. In this case, the heuristic behind is to replace the non adequate term by a new variable (therefore, generalizing the query) but restricting that variable with features of the replaced term (that is to say, triples in the source dataset which the replaced term is the subject). The expression for such a rule is the following:

```
PREFIX s:<source dataset>
PREFIX t:<target dataset>
REPLACE #s #p s:u
BY      #s #p ?v .
        AND(?v #f #o)
WHEN {
      s:u #f #o  }
```

The algorithm $\mathcal{A}$ devised for applying the rewriting rules consists in applying first those rules that seem to maintain as much as possible the semantics of the current query. The rewriting algorithm applies the rules on a kind by kind basis. Within a kind of rules the algorithm repeats the application of each rule until no more application is possible. The rules of a kind are sequentially numbered and they are applied in that numbered sequence. A rule is applied as long as its preconditions (described by columns *REPLACE* and *WHEN*) are satisfied. When a rule is no longer applicable, the algorithm drives to the following rule. Something specific takes place when application of a feature-based rule has finished. If any non adequate IRI remains in the query, Equivalence and Hierarchy rules are tried again and after that, any triple pattern presenting a non adequate IRI is deleted from the query. At any moment the current query being object of rewriting becomes adequate for the target dataset, the algorithm stops and return such target query. Next, a more explicit description of the algorithm is presented, where Q_s and Q_t represent the source and target query, respectively, and $C = D_s \cup D_t \cup D_b$ represent the data graph context, composed of the source, target and bridge datasets, where the rewriting process takes place. $voc(Q_t)$ and $voc(D_t)$ respectively mean the vocabulary (i.e. set of terms) of Q_t and D_t.

REWRITE(Q_s, C)
// $C = D_s \cup D_t \cup D_b$
$Q_t \leftarrow Q_s$
if $voc(Q_t) \nsubseteq voc(D_t)$ **then**
  $Q_t \leftarrow$APPLY(EquivalenceRules, Q_t, C)
if $voc(Q_t) \nsubseteq voc(D_t)$ **then**
  $Q_t \leftarrow$APPLY(HierarchyRules, Q_t, C)
if $voc(Q_t) \nsubseteq voc(D_t)$ **then**

$Q_t \leftarrow$APPLY(AnswerBasedRules, Q_t, C)
if $voc(Q_t) \nsubseteq voc(D_t)$ **then**
$Q_t \leftarrow$APPLY(ProfileBasedRules, Q_t, C)
if $voc(Q_t) \nsubseteq voc(D_t)$ **then**
$Q_t \leftarrow$APPLY(FeatureBasedRules, Q_t, C)
if $voc(Q_t) \nsubseteq voc(D_t)$ **then**
$Q_t \leftarrow$APPLY(EquivalenceRules, Q_t, C)
if $voc(Q_t) \nsubseteq voc(D_t)$ **then**
$Q_t \leftarrow$APPLY(HierarchyRules, Q_t, C)
$Q_t \leftarrow$ deleteNonAdequateTriplePatterns($Q_t, voc(D_t)$)
return Q_t

APPLY(*RuleSet*, Q, C)
for each $r \in$ *RuleSet*
while *applicable*(r, Q, C)
$Q \leftarrow$ *rewriteWith*(r, Q, C)
return Q

The similarity factor $\mathcal{SF}$ associated to a target query is an aggregation of the similarity values associated to each rule applied to reach such a target query. Among different possibilities, a measure based on the Euclidean distance on a n-dimensional space was selected. Given a sequence of rule applications $(r_i)_{i=1}^N$ for the rewriting of a source query into a target query, involving the corresponding non adequate terms $(u_i)_{i=1}^N$, the values $(\phi(u_i))_{i=1}^N$ can be considered the coordinates of a point in a N-dimensional space, where the point $(1, \ldots, 1)$ represents the best and the point $(0, \ldots, 0)$ the worst. Then, the Euclidean distance between the points $(\phi(u_i))_{i=1}^N$ and $(1, \ldots, 1)$ provides a foundation for a similarity measure. In order to normalize the similarity value within the real interval $[0, 1]$, with the value 1 representing the best similarity, the Euclidean distance between $(\phi(u_i))_{i=1}^N$ and $(1, \ldots, 1)$ is divided by $\sqrt{N}$, and substracted from the best similarity 1. Let us use $(u_i)_{i=1}^N$ in representation of the sequence of rules $(r_i)_{i=1}^N$, then

$$\mathcal{SF}((u_i)_{i=1}^N) = 1 - \frac{1}{\sqrt{N}} \sqrt{\sum_{i=1}^{N} (1 - \phi(u_i))^2}$$

Finally, the score selected to inform about the quality of the obtained target query was the F1 score calculated by comparing the answers retrieved by the target query with those retrieved by the corresponding Gold standard query. We call *Relevant answers* (Rel) to the set of answers obtained by running the Gold standard query, and *Retrieved answers* (Ret) to the set of answers obtained by running the target query. Then, the values Precision (P), Recall (R), and F1 score (F1) are calculated with the following formulae.

$$\mathrm{P} = \frac{|\mathrm{Rel} \cap \mathrm{Ret}|}{|\mathrm{Ret}|} \qquad \mathrm{R} = \frac{|\mathrm{Rel} \cap \mathrm{Ret}|}{|\mathrm{Rel}|} \qquad \mathrm{F1} = 2 \times \frac{\mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}}$$

The supervised learning model $\mathcal{P}$ devised to predict the F1 score of a target query was generated from the application of a Random Forest algorithm. Some other regression algorithms were considered and after an experimentation process, discussed in section B.5, the Random Forest was selected because it offered the best results.

B.5 Framework validation

This section presents the main results of the process carried out to validate the proposed framework. The following resources are presented: (a) the LOD datasets selected for querying data, (b) the collection of training queries, (c) the optimization algorithm used to determine the proper parameter values for computing similarity measures and a discussion of its results, (d) the collection of features gathered from data to construct the learning datasets used by the machine learning algorithms and the results obtained by them, (e) the processing times needed by the framework implementation to get the answers in the corresponding SPARQL endpoints.

B.5.1 Datasets and queries

To validate the framework we trusted on well known datasets of the Linked Open Data environment, with accessible endpoints that facilitate the assessment of our experiments. Three domain areas were considered for the datasets: media-domain, bibliographic, and life science. For each one, a set of recognized datasets were selected. With respect to media-domain, the selected ones were: DBpedia, MusicBrainz, LinkedMDB, Jamendo, New York Times and BBC. With respect to bibliographic domain, we considered BNE (Biblioteca Nacional de España), BNF (Bibliothèque National du France), BNB (British National Bibliography), LIBRIS, and Cambridge. And finally for the life science area: Drugbank, SIDER, CHEBI, DISEASOME, and KEGG were the selected ones. Moreover, to achieve greater plurality in the tests, we used the SP2Bench, which is based on a synthetic dataset. In addition, our framework implementation also considered DBpedia, VIAF, Freebase, and GeoNames as bridge datasets. The available SPARQL endpoint for each mentioned dataset was used to answer the queries. In summary, we considered 17 different datasets (16 real + 1 synthetic) along with 4 bridge datasets that assisted us on the framework implementation. This is a evidence of the broad coverage of the experiments performed.

The set of experimental queries were selected after analyzing heterogeneous benchmarks such as QALD[5], FedBench [SGH+11a], and real SPARQL endpoint logs (like BNE or DBpedia). A set of 100 queries was created for experimenting with the framework and providing data for the learning process. Those queries along with their corresponding gold standards and

[5] https://qald.sebastianwalter.org/

the names of source and target datasets are listed in the appendix published in [TB18b, TB18a]. The idea underlying the selection process was to select queries that could be representative of the different SPARQL query types and that could cover heterogeneous domains in the Linked Open Data framework. Concerning provenance we selected 25 queries from well known cited benchmarks that were defined for some of the datasets listed in the previous paragraphs, and that presented a variety of graph pattern structures. Furthermore, 25 more queries were selected from the LOD SPARQL endpoints logs (year 2014 period). To select them, a clustering of the queries was carried out according to their graph pattern structure, and a random sample of each group was chosen, previously eliminating those queries that were malformed or those that exceed a maximum of 15 triple patterns, since they are usually triple pattern repetitions that do not provide structural diversity to the query set. In this way we got an initial set of 50 queries, which we doubled by converting their gold standards into source queries that were the origin of a new rewriting. In total we obtained a set of 100 queries expressed in terms of vocabularies of 17 different datasets, accessible via SPARQL endpoints.

Regarding the syntactic structure of the queries, a variety of the SPARQL operators (UNION, OPTIONAL, FILTER) and different patterns for joins of variables appeared in the queries. The number of triple patterns of each query ranges from 1 to 7.

A very important source of knowledge that supports this framework are repositories containing mappings between terms from different vocabularies and interlinkings between different IRIs for the same resource. For instance, we can find files with mapping triples in the DBpedia project[6] or data dumps of Freebase/Wikidata mappings. Some datasets, for instance Jamendo[7], are accompanied by sets of mappings that interlink their resources with resources in another domain sharing dataset, like Geonames and Musicbrainz. Moreover, some datasets can act as a central point of interlinking between some different datasets. It is the case of VIAF[8] on the bibliographic domain. A very useful web service, helping with this problem is http://sameas.org/, a service that helps to solve the existence of co-references between different data sets. But unfortunately there are no many well organized repositories and services of this kind, For this reason, to improve our approach we have created our own mapping repository in a local instance using Virtuoso Open Source triple store. So we have crawled the web finding possible mapping files and have incorporated them into the Virtuoso repository.

B.5.2 Suitability of the similarity factor

The similarity factor $\mathcal{SF}$ in the framework is intended to inform the user about a similarity estimation of a target query with respect to a source query. Our approach takes into account the query content and the query

[6] http://wiki.dbpedia.org/services-resources/interlinking
[7] http://dbtune.org/jamendo/
[8] http://viaf.org/

results as dimensions for query similarity, and its computation is based on a kind of graph-edit distance associated to each rewriting rule application. Assuming that the ideal for the target query would be to behave as similarly as possible to the corresponding gold standard query, along the query results dimension, it is natural to design $\mathcal{SF}$ in such a way that the similarity factor associated to a target query be correlated with the F1 score of that target query. Remember that such F1 score is computed for the target query with respect to the predefined gold standard query. Therefore, tuning of the similarity measures used to compute $\mathcal{SF}$ is desirable.

The similarity measure, presented in section B.4, is based on similarity functions ϕ (associated to each rule application $\mathcal{V}(r)$) and many of them are defined as a linear combination of three similarity measures (S_n, S_d, S_o), involving three parameters α_n, α_d, and α_o. Instead of trying to determine their appropriate values by chance it seems preferable to devise a method to optimize their values towards the goal of moving $\mathcal{SF}$ closer to F1 score. This is the tuning process to which we refer in the previous paragraph.

We selected a method based on a genetic algorithm, specifically the Harmony Search (HS) algorithm [GKL01]. Harmonies represent sets of variables to optimize, whereas the quality of the harmony is given by the fitness function of the optimization problem at hand.

In our case the variables to optimize are the parameters $(\alpha_n, \alpha_d, \alpha_o)$ appearing in the definition of the similarity measure. And the established fitness function was the maximization of the proportion of m queries whose absolute difference between the value of the similarity factor $\mathcal{SF}(q_i)$ and the F1 score for query number $i = 1 \ldots m$ was smaller than a given threshold β. The fitness function is as follows:

$$\begin{aligned}
&\text{maximize} \quad \sum_{i=1}^{m} \frac{1}{m} H(q_i)) \\
&\text{subject to} \quad 0 \leq \alpha_n, \alpha_d, \alpha_o, \beta \leq 1 \\
&\text{where} \\
&H(q_i) = \begin{cases} 1 & \text{if } |F_1(q_i) - \mathcal{SF}(q_i, \alpha_n, \alpha_d, \alpha_o)| < \beta \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

In order to carry out the optimization process the query set (see B.5.1) was considered, and five-fold cross validation were performed. The sample data (initial query set) are divided into five subsets. One of the subsets is used as test data and the other four as training data. The cross-validation process is repeated five times (folds), with each of the five subsamples used exactly once as the test data. For each iteration (fold), the first step was to execute the HS algorithm on the set of training queries (composed by four subsets), in order to obtain the parameter values that achieve optimal fitness. For this, the algorithm was parametrized with the number of iterations and initial values for the parameters. The HS optimization process may obtain different solutions depending on the initial random values chosen for the parameters and the number of iterations allowed.

FIGURE B.1: Convergence of fitness with the training dataset and $\beta = 0.2$.

With 100 iterations and an initialization defined by the HS algorithm itself, we obtained the parameter values shown in table B.3, according to different values of β $(0.4, 0.2, 0.1)$. Each value shown in the cells of the table represents the different results obtained for training subsamples in each iteration (fold).

	0.1					**0.2**					**0.4**				
	1-fold	**2-fold**	**3-fold**	**4-fold**	**5-fold**	**1-fold**	**2-fold**	**3-fold**	**4-fold**	**5-fold**	**1-fold**	**2-fold**	**3-fold**	**4-fold**	**5-fold**
α_n	0.119	0.120	0.118	0.112	0.114	0.130	0.130	0.129	0.129	0.130	0.134	0.130	0.131	0.137	0.134
α_d	0.501	0.504	0.500	0.501	0.504	0.517	0.514	0.513	0.519	0.515	0.537	0.529	0.534	0.535	0.538
α_o	0.379	0.369	0.379	0.379	0.375	0.351	0.351	0.352	0.354	0.353	0.328	0.327	0.329	0.319	0.321

TABLE B.3: Optimal parameter values for similarity function calculated from training subsamples for each fold.

One example of convergence of the HS optimization process for the training dataset and $\beta = 0.2$ is shown in figure B.1, where abscissas axis represents the number of algorithm iterations and the ordinate axis represents the fitness value. It can be observed how the fitness increases with the number of iterations.

To assess the validity of the parameter values obtained by the algorithm, the similarity factor was computed for each fold over the set of remaining test queries (in this case using the alphas obtained in the different scenarios with $\beta = 0.4, 0.2, 0.1$, respectively) and then, the absolute difference between these similarity factors and the F1 scores for the corresponding target queries were calculated. Training and Test Fitness in table B.4 display the fitness mean values for the five different folds calculated over the training dataset and test dataset respectively, along with the corresponding Mean Absolute Error (MAE). The MAE is the difference between the training dataset fitness value and the one obtained with the test dataset; it measures the suitability of the optimization. It can be observed that the MAE never exceeds 0.15, which indicates that the optimization process is valid (values less than 0.3 are considered valid).

Taking into account that a tighter threshold as $\beta = 0.1$ produces worse test fitness values and higher absolute errors, and a looser threshold as $\beta = 0.4$ is too relaxed for similarity considerations, we decided to implement the

computation of $\mathcal{SF}$ with the values $\alpha_n = 0.130$, $\alpha_d = 0.515$, and $\alpha_o = 0.352$, corresponding to the threshold $\beta = 0.2$, which offers a reasonable balance between test fitness values and closeness of $\mathcal{SF}$ and F1.

TABLE B.4: Fitness values for different thresholds.

	Training Fitness	*Test Fitness*	*Mean Absolute Error (MAE)*
β **= 0.4**	0.832	0.761	0.071
β **= 0.2**	0.809	0.725	0.084
β **= 0.1**	0.678	0.532	0.146

B.5.3 Discussion

In the following we discuss the validity of the similarity factor $\mathcal{SF}$ obtained by our embodied framework using the optimized parameter values calculated by the HS aforementioned method over the set of 100 experimental queries. In order to trust in the quality of the information conveyed to the user by the similarity factor, it is relevant to compare such factor with a measure of the behaviour of the target query. It is evident that the F1 score is a measure of that kind and therefore we proceeded with such a comparison.

Figure B.2 shows a scatter plot of the 100 points with coordinates (F1 score, $\mathcal{SF}$) and table B.5 shows numbers for the same points. An analysis of the results revealed the following considerations: In 59 out of 100 source queries the target queries provided the same set of results as the corresponding gold standard queries. From this set, in 50 of them their F1 score was 1, and in 9 of them (Q11, Q17, Q25, Q26, Q39, Q41, Q52, Q76, and Q91) the F1 score could not be calculated because the sets of relevant and retrieved results (see section B.4) were both empty (notice that eventual dataset updates could change those results). The cases in which the F1 score equals 1 (50% of the whole set) can be divided into two groups depending on the similarity factor: (1) Cases whose similarity factor equals F1 score, represent a 23% of the whole query set. (2) Cases whose similarity factor is less than F1 score, represent a 27% of the whole query set. In the other case, there were 41 queries in which the target queries did not provide the same set of results as the corresponding gold standard queries. Therefore these queries had a F1 score lower than 1. From this set, in 14 of them the similarity factor was lower than the F1 score. We want to highlight that in cases where $\mathcal{SF}$ < F1 score, the rewriting system is performing better than the offered similarity factor, since the higher F1 score shows that the target query performs more similarly to the gold standard than the offered information.

Finally, in 25 of them the similarity factor was higher than the F1 score. In those cases the similarity factor was too optimistic because the actual results provided by the target query were quite different from those provided by the gold standard query and there were cases where the F1-measure value was very low. This circumstance support the idea that would be interesting to complement the information to the user with a

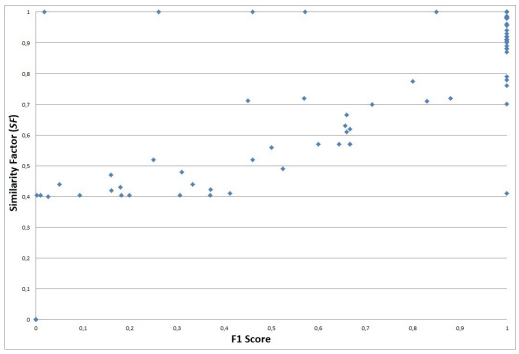


FIGURE B.2: Scatterplot for F1 score and Similarity factor (using similarity parameter values calculated for training dataset and $\beta = 0.2$).

quality score reflecting the F1-score. As long as gold standard queries are not present in a real scenario, devising a prediction model for such a score is an option. Next, section B.5.4 will explain an implementation of such a predictive model. There were also 2 queries where the retrieved results were empty (Q28, Q40).

Concluding this comparison, it can be said that $\mathcal{SF}$ is a cautious information to the user since in the majority of the cases $\mathcal{SF} \leq$ F1, and therefore frequently indicates a lower bound quality of the behaviour of the target query with respect to expected answers. It is interesting to remark that while $\mathcal{SF}$ reflects an intensional measure (semantic similarity of the replacement), the F1 score has an extensional character.

As discussed above, three domain areas were considered for the datasets: media-domain, bibliographic, and life science. Table B.7 presents some statistics about the queries distribution and their $\mathcal{SF}$ and F1 values. In particular, the number of queries per domain, their similarity factor and F1 score averages ($\overline{\mathcal{SF}}$, $\overline{F1}$) and their corresponding standard deviations (σ).

In that table B.7 we can observe that $\overline{\mathcal{SF}}$ and $\overline{F1}$ values do not vary significantly depending on the domain. The greatest distances are, in the case of $\overline{\mathcal{SF}}$, between Media and Life Science domains (0.073) and, in the case of $\overline{F1}$, between Media and Bibliographic domain (0.07), never exceeding a difference greater than 0.08. Moreover, Life Science is the domain in which the values are more dispersed with relation to the average, which means that the quality of the rewriting, even within the same domain, has varied significantly. The best results are obtained in the Media domain, due to a greater number of links among source, target and bridge datasets belonging to that domain. Nevertheless, we are aware that the limited number of queries considered in the testbed may have an impact in the compared behaviour of the respective domains. However, the obtained results are quite promising and therefore they could be considered as a baseline.

Finally, we found interesting to know what was the correlation between

TABLE B.5: $\mathcal{SF}$ (using similarity parameter values calculated for training dataset and $\beta = 0.2$) and F1 score for the experimental query set.

	Q1	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**
F1	1	0.002	1	1	0.83	1	0.46	1	0.5	1
$\mathcal{SF}$	0.956	0.405	0.987	0.979	0.71	0.905	0.52	0.984	0.56	0.78
	Q11	**Q12**	**Q13**	**Q14**	**Q15**	**Q16**	**Q17**	**Q18**	**Q19**	**Q20**
F1	-	1	1	1	1	1	-	1	1	1
$\mathcal{SF}$	-	0.912	1	1	1	1	-	0.901	1	1
	Q21	**Q22**	**Q23**	**Q24**	**Q25**	**Q26**	**Q27**	**Q28**	**Q29**	**Q30**
F1	1	0.66	0.16	0.05	-	-	0.01	-	0.8	1
$\mathcal{SF}$	1	0.61	0.47	0.44	-	-	0.405	-	0.775	0.701
	Q31	**Q32**	**Q33**	**Q34**	**Q35**	**Q36**	**Q37**	**Q38**	**Q39**	**Q40**
F1	1	1	1	0.57	0.88	1	0.18	1	-	-
$\mathcal{SF}$	1	0.79	0.87	0.72	0.72	0.93	0.43	1	-	-
	Q41	**Q42**	**Q43**	**Q44**	**Q45**	**Q46**	**Q47**	**Q48**	**Q49**	**Q50**
F1	-	1	1	0.31	0.37	0.25	1	1	1	0.45
$\mathcal{SF}$	-	0.88	0.41	0.48	0,405	0.52	1	1	0.761	0.711
	Q51	**Q52**	**Q53**	**Q54**	**Q55**	**Q56**	**Q57**	**Q58**	**Q59**	**Q60**
F1	1	-	0.666	1	0.018	0.198	0.666	1	1	1
$\mathcal{SF}$	0.96	-	0.57	0.94	1	0.405	0.665	0.919	0.982	0,89
	Q61	**Q62**	**Q63**	**Q64**	**Q65**	**Q66**	**Q67**	**Q68**	**Q69**	**Q70**
F1	1	0.371	0.306	0.656	0.666	0.714	1	1	1	1
$\mathcal{SF}$	1	0.422	0.405	0.63	0.62	0.7	1	0.88	1	1
	Q71	**Q72**	**Q73**	**Q74**	**Q75**	**Q76**	**Q77**	**Q78**	**Q79**	**Q80**
F1	1	0.666	0.571	0.26	1	-	0.85	1	0.46	1
$\mathcal{SF}$	1	0.57	1	0.99	1	-	1	1	1	1
	Q81	**Q82**	**Q83**	**Q84**	**Q85**	**Q86**	**Q87**	**Q88**	**Q89**	**Q90**
F1	1	1	1	0.026	0.644	0.412	0.181	1	1	0.6
$\mathcal{SF}$	1	0.98	0.91	0.4	0.57	0.41	0.405	1	0.96	0.57
	Q91	**Q92**	**Q93**	**Q94**	**Q95**	**Q96**	**Q97**	**Q98**	**Q99**	**Q100**
F1	-	1	1	0.524	0.093	0.333	1	1	1	0.16
$\mathcal{SF}$	-	0.92	1	0.49	0.405	0.44	1	0.98	0.91	0.42

TABLE B.6: Summary of the comparison between $\mathcal{SF}$ value and F1 score.

<table>
<tr><td colspan="6">Query set composed of
100 queries</td></tr>
<tr><td colspan="3">59 queries with
Retrieved answers = Relevant answers</td><td colspan="3">41 queries with
Ret≠ Rel</td></tr>
<tr><td colspan="2">50 queries with
F1 = 1</td><td rowspan="2">9 queries
with |Ret|=|Rel|=0
F1 cannot
be calculated</td><td rowspan="2">25 queries with
$\mathcal{SF}$ > F1</td><td rowspan="2">14 queries with
$\mathcal{SF}$ < F1</td><td rowspan="2">2 queries with
|Ret|=0</td></tr>
<tr><td>23 queries with
$\mathcal{SF}$ = F1</td><td>27 queries with
$\mathcal{SF}$ < F1</td></tr>
</table>

TABLE B.7: $\mathcal{SF}$ and F1 averages with standard deviation

	N° Queries	$\overline{\mathcal{SF}}$ - σ	$\overline{F1}$ - σ
Media domain	34	0.707 - 0.37	0.729 - 0.31
Bibliographic domain	40	0.649 - 0.4	0.659 - 0.33
Life Science domain	26	0.634 - 0.47	0.72 - 0.38

TABLE B.8: Selected features for the 8 different datasets.

		F.1	F.2	F.3	F.4	F.5	F.6	F.7	F.8
1	Equivalence rule application times	X	X						
2	Equivalence rule similarity measure value	X	X	X					
3	Hierarchy rule application times	X	X						
4	Hierarchy rule similarity measure value	X	X	X					
5	Answer rule application times	X	X						
6	Answer rule similarity measure value	X	X	X					
7	Profile rule application times	X	X						
8	Profile rule similarity measure value	X	X	X					
9	Feature rule application times	X	X						
10	Feature rule similarity measure value	X	X	X					
11	Similarity factor	X	X	X	X	X	X	X	X
12	Number of source triple patterns	X	X						
13	Number of terms in source query	X	X	X	X	X	X	X	
14	Number of non-adequate terms	X	X	X	X	X	X		X
15	Number of union operators	X							
16	Number of projected variables	X							
17	Number of optional operators	X							
18	Number of filter operators	X							
19	Source Dataset	X	X	X	X				
20	Target Dataset	X	X	X	X				
21	Number of mappings between source and target	X	X			X			

TABLE B.9: R2 metric of the predictive models.

Datasets	**LR**	**SVM**	**RF**
Features1	0.6751	-1.5072	**0.8219**
Features2	0.5902	-1.0313	0.7132
Features3	0.6045	-0.0437	0.7152
Features4	0.4572	-0.0689	0.7026
Features5	0.7146	**0.7731**	0.6835
Features6	**0.7529**	0.6815	0.7797
Features7	0.7511	0.7611	0.7221
Features8	0.7523	0.7146	0.7923

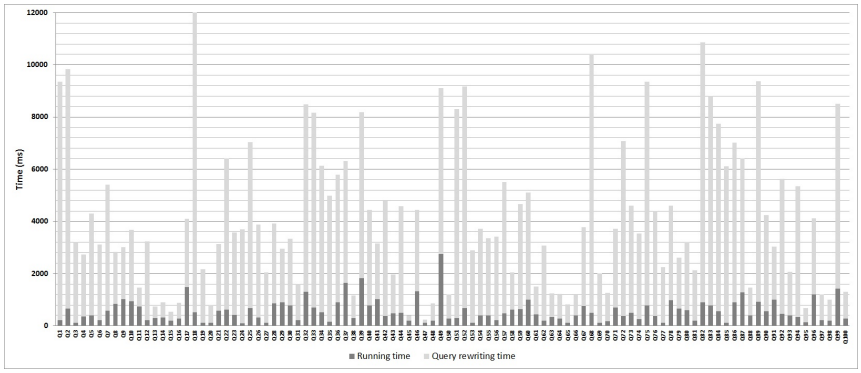


FIGURE B.3: Answering times plus rewriting times.

TABLE B.10: Processing times in *ms*

Query	**TT**	**AT**	**TAT**	**Query**	**TT**	**AT**	**TAT**
Q1	8926	214	9140	Q51	7699	307	8006
Q2	8525	652	9177	Q52	7836	672	8508
Q3	2991	108	3099	Q53	2643	121	2764
Q4	2005	364	2369	Q54	2893	408	3301
Q5	3500	405	3905	Q55	2560	396	2956
Q6	2688	210	2898	Q56	2971	221	3192
Q7	4261	573	4834	Q57	4562	475	5037
Q8	1142	847	1989	Q58	797	623	1420
Q9	981	1021	2002	Q59	3400	637	4037
Q10	1802	938	2740	Q60	3111	1002	4113
Q11	0	734	734	Q61	656	429	1085
Q12	2825	208	3033	Q62	2676	198	2874
Q13	167	290	457	Q63	581	335	916
Q14	268	312	580	Q64	671	281	952
Q15	137	205	342	Q65	592	112	704
Q16	330	277	607	Q66	407	392	799
Q17	1144	1482	2626	Q67	2273	752	3025
Q18	11264	523	11787	Q68	9380	503	9883
Q19	1953	109	2062	Q69	1774	122	1896
Q20	536	108	644	Q70	921	173	1094
Q21	1994	574	2568	Q71	2327	696	3023
Q22	5191	613	5804	Q72	6317	384	6701
Q23	2753	409	3162	Q73	3612	493	4105
Q24	3503	102	3605	Q74	3037	251	3288
Q25	5658	689	6347	Q75	7780	782	8562
Q26	3258	314	3572	Q76	3649	373	4022
Q27	1796	125	1921	Q77	2030	109	2139
Q28	2225	852	3077	Q78	2653	974	3627
Q29	1153	901	2054	Q79	1264	670	1934
Q30	1765	782	2547	Q80	1983	593	2576
Q31	1181	213	1394	Q81	1727	201	1928
Q32	5899	1297	7196	Q82	9052	904	9956
Q33	6749	708	7457	Q83	7264	775	8039
Q34	5096	514	5610	Q84	6613	562	7175
Q35	4658	160	4818	Q85	5883	118	6001
Q36	4005	898	4903	Q86	5216	905	6121
Q37	3013	1650	4663	Q87	3840	1284	5124
Q38	571	294	865	Q88	685	394	1079
Q39	4512	1834	6346	Q89	7523	928	8451
Q40	2899	775	3674	Q90	3137	551	3688
Q41	1103	1028	2131	Q91	1022	1005	2027
Q42	4020	383	4403	Q92	4705	462	5167
Q43	1029	469	1498	Q93	1280	392	1672
Q44	3598	493	4091	Q94	4659	347	5006
Q45	0	197	197	Q95	389	144	533
Q46	1779	1328	3107	Q96	1711	1206	2917
Q47	0	122	122	Q97	775	215	990
Q48	461	203	664	Q98	621	193	814
Q49	3613	2751	6364	Q99	5652	1431	7083
Q50	653	272	925	Q100	739	285	1024

the computed similarity factor and the F1 score. The Pearson correlation coefficient (usually named Pearson's r) is a popular measure of the strength and direction of the linear relationship between two variables. Pearson's correlation coefficient for continuous data ranges from −1 to +1. A value equals to 0 indicates no linear relationship between the variables. Positive correlation indicates that both variables increase or decrease together, whereas negative correlation indicates that as one variable increases, so the other decreases, and vice versa. In our case, the value was $r = 0.724$, which can be considered a high positive correlation, indicating that variables (F1 score, $\mathcal{SF}$) increase or decrease together providing a coherent metric for informing the user about the outcome of the rewriting process.

We think that the results in table B.5 allow us to say that the similarity factor defined in section B.4 is quite informative about the quality of the target query from an intensional point of view. Nevertheless, one goal of the presented framework is to serve as a tool for establishing benchmarks which promote improvement of query rewriting systems, and the embodiment presented in this paper could be considered a baseline.

B.5.4 Predictive model for the F1 score

As we have already mentioned, gold standard queries are not available in a real scenario, that is the reason why a predictive model $\mathcal{P}$ was considered in the framework. In the scenario of this paper, $\mathcal{P}$ is in charge of predicting the F1 score. Such predicted value is the *quality score*, referred in the example shown in section B.1, that adds information to the user.

The construction of that predictive model was based on learning datasets that contain features of data that represent underlying structure and characteristics of the data subject of the prediction. In our scenario, features related to the structure of the source query such as number of triple patterns or number of operators, along with features related to rules that take part during the rewriting process, and finally features concerning the involved LOD datasets, were considered to build the feature datasets.

Following we present the 21 considered features:

1. Similarity and rules features, numbered from 1 to 11: (1) Number of times the equivalence rules are applied, (2) Similarity measure value associated to the equivalence rules application, (3) Number of times the hierarchy rules are applied, (4) Similarity measure value associated to the hierarchy rules application, (5) Number of times the answer-based rules are applied, (6) Similarity measure value associated to the answer-based rules application, (7) Number of times the profile-based rules are applied, (8) Similarity measure value associated to the profile-based rules application, (9) Number of times the feature-based rules are applied, (10) Similarity measure value associated to the feature-based rules application, and (11) the similarity factor calculated for the target query.

2. Query structure features, numbered from 12 to 18: (12) Number of triple patterns of the source query, (13) number of terms of the source

query, (14) number of non adequate terms for the target dataset, (15) number of union operators, (16) number of projected variables, (17) number of optional operators, and (18) number of filter operators.

3. LOD Datasets features, numbered from 19 to 21: (19) categorical data associated to the source dataset depending on its size. Three values are possible: 1, for small datasets with less than 10^5 triples; 2, for medium size datasets with a number of triples between 10^5 and 10^6; 3 for larger datasets with more than 10^6 triples, (20) categorical data associated to the target dataset depending on its size (the same possible values as in the case of feature number 20), and (21) number of mappings between source and target datasets.

In order to select a best fit model, we experimented with the following off-the-shelf algorithms [MRT12, WEG87]: Linear regression (LR), Support Vector Machines (SVM), and Random Forest; and with 8 different datasets (F.1 to F.8) corresponding to distinct feature selection (see table B.8).

The values used in the experiment were obtained from the rewriting of the 100 aforementioned queries. This set of queries were divided in three fragments: 80% for the training process, 15% for the validation process, and 5% for the test process, respectively. The score of each of those models was measured based on a 20-fold cross-validated average mean squared error-R2 metric. The results are presented in the table B.9, where each cell of the table represents the coefficient of determination for each model trained with the feature dataset indicated by the row.

As can be seen, the model that best fit is that obtained using the Random Forest-RF algorithm with F.1 dataset, with a R2 equals to 0.8219 for validation set. Moreover, notice that the features datasets F.6, F.7, and F.8 are the ones that, in general, show a better behaviour with all the models. Therefore, the similarity factor (11), number of terms (13), and number of non-adequate terms (14) features can be considered the most significant ones. And it points out again the validity of the computed similarity factor. To asses the generalization error of the final chosen model, the value of R2 over the test set was computed, yielding a value of 0.8014.

B.5.5 Processing time

The framework is placed into the Linked Open Data environment, leveraging datasets SPARQL endpoints. In order to asses the performance of the framework we want to show runtimes and the evaluation conditions in which it was implemented.

The rewriting process (rules and algorithm) has been implemented with Attributed Graph Grammar System (AGG)[Tae04]. For similarity computation, we relied on the libraries Wordnet Similarity for Java (WS4J)[9] and SimMetrics[10]. The queries run by means of Jena semantic framework[11].

[9] https://code.google.com/p/ws4j/
[10] http://sourceforge.net/projects/simmetrics/
[11] https://jena.apache.org/

For performance testing, the system consisted of an Intel Core 2 Duo 2.67 GHz processor, 8 GB RAM, Windows 7 Professional, and Java Runtime Environment 1.8. All measurements were executed six times consecutively using the average of the last five measurements.

Table B.10 displays the processing times for each query in benchmark executed over the corresponding dataset. The TT column indicates the time needed by the framework to obtain the rewritten query. In column AT the time to get the answers of the query over the SPARQL endpoint is indicated, and finally the column TAT is the sum of the previous times (TT+AT). The same information is graphically showed in Figure B.3. The TT times, that really represents the performance of our system, are between a minimum value of $137ms$ (Q15) and a maximum value of $11264ms$ (Q18), regardless of the values of $0ms$ (Q45, Q47). And 90% of queries are executed in a maximum time of $6sec$. Taking into account this significant performance information about the framework implementation, we consider that the processing times are acceptable but amenable to improvement.

B.6 Conclusions

The current state of the Web of Data with so many different datasets of a heterogeneous nature makes it difficult for users to query those datasets in order to exploit the vast amount of data they contain. Different proposals are appearing to overcome that limitation. In this paper we have detailed the features of a framework that allows end users to obtain results from different datasets expressing the query using only the vocabulary which the users are more familiar with, and informs them about the quality of the answer. Moreover, this framework serves technical users as a tool for establishing query rewriting benchmarks.

The framework has been embodied with a selected set of rules, rule scheduling algorithm, similarity measures, and quality estimation model composed of similarity factor function and F1 score predictive model. Moreover, the framework has been validated in a real scenario and the results obtained are promising, and they could be considered a baseline to be improved considering smarter rewriting rules and better shaped similarity measures.

Bibliografía

[ABD+12] Sören Auer, Lorenz Bühmann, Christian Dirschl, Orri Erling, Michael Hausenblas, Robert Isele, Jens Lehmann, Michael Martin, Pablo N Mendes, Bert Van Nuffelen, et al. Managing the life-cycle of linked data with the lod2 stack. In *International semantic Web conference*, pages 1–16. Springer, 2012.

[AFMPdlF11] Mario Arias, Javier D Fernández, Miguel A Martínez-Prieto, and Pablo de la Fuente. An empirical study of real-world sparql queries. *Proc. 1st International Workshop on Usage Analysis and the Web of Data, USEWOD*, 2011.

[AH09] Keith Alexander and Michael Hausenblas. Describing linked datasets-on the design and usage of void, the'vocabulary of interlinked datasets. In *In Linked Data on the Web Workshop (LDOW 09), in conjunction with 18th International World Wide Web Conference (WWW 09*. Citeseer, 2009.

[AVL+11] Maribel Acosta, Maria-Esther Vidal, Tomas Lampo, Julio Castillo, and Edna Ruckhaus. Anapsid: an adaptive query processing engine for sparql endpoints. In *International Semantic Web Conference*, pages 18–34. Springer, 2011.

[BHBL09] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[Biz09] Christian Bizer. The emerging web of linked data. *IEEE intelligent systems*, -(5):87–92, 2009.

[BL13] Tim Berners-Lee. Giant global graph, november 2007. *Available from World Wide Web*, 2013.

[BLBC+04] Tim Berners-Lee, Tim Bray, Dan Connolly, Paul Cotton, Roy Fielding, Mario Jeckle, Chris Lilley, Noah Mendelsohn, David Orchard, Norman Walsh, et al. Architecture of the world wide web, volume one. *version*, 20041215:W3C, 2004.

[BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):34–43, 2001.

[Bri04] Dan Brickley. Rdf vocabulary description language 1.0: Rdf schema. *http://www. w3. org/TR/rdf-schema/*, 2004.

[C+12] World Wide Web Consortium et al. R2rml: Rdb to rdf mapping language. *W3C Working Draft*, 2012.

[CCK+17] Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao. Ontop: Answering SPARQL queries over relational databases. *Semantic Web*, 8(3):471–487, 2017.

[CFMV11] Silvana Castano, Alfio Ferrara, Stefano Montanelli, and Gaia Varese. Ontology and instance matching. In *Knowledge-driven multimedia information extraction and ontology evolution*, pages 167–195. Springer, 2011.

[CH14] Michelle Cheatham and Pascal Hitzler. The properties of property alignment. In *Proceedings of the 9th International Conference on Ontology Matching-Volume 1317*, pages 13–24. CEUR-WS. org, 2014.

[CSM+10a] Gianluca Correndo, Manuel Salvadores, Ian Millard, Hugh Glaser, and Nigel Shadbolt. Sparql query rewriting for implementing data integration over linked data. In *Proceedings of the 2010 EDBT/ICDT Workshops*, EDBT '10, pages 4:1–4:11, New York, NY, USA, 2010. ACM.

[CSM+10b] Gianluca Correndo, Manuel Salvadores, Ian Millard, Hugh Glaser, and Nigel Shadbolt. Sparql query rewriting for implementing data integration over linked data. In *Proceedings of the 2010 EDBT/ICDT Workshops*, EDBT '10, pages 4:1–4:11, New York, NY, USA, 2010. ACM.

[CWL14] Richard Cyganiak, David Wood, and Markus Lanthaler. *RDF 1.1 Concepts and Abstract Sysntax*. World Wide Web Consortium, 2014. W3C Recommendation 25 February 2014.

[DAB16] Gonzalo Diaz, Marcelo Arenas, and Michael Benedikt. Sparqlbye: Querying rdf data by example. *Proceedings of the VLDB Endowment*, 9(13):1533–1536, 2016.

[DESTdS11] Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The alignment api 4.0. *Semantic Web*, 2(1):3–10, 2011.

[DG13] Renata Dividino and Gerd Gröner. Which of the following sparql queries are similar? why? In *Proceedings of the First International Conference on Linked Data for Information Extraction - Volume 1057*, LD4IE'13, pages 2–13, Aachen, Germany, Germany, 2013. CEUR-WS.org.

[DLSM17] Dennis Diefenbach, Vanessa Lopez, Kamal Singh, and Pierre Maret. Core techniques of question answering systems over knowledge bases: a survey. *Knowledge and Information systems*, pages 1–41, 2017.

[DSW06] Peter Dolog, Heiner Stuckenschmidt, and Holger Wache. Robust query processing for personalized information access on the semantic web. In *Proceedings of the 7th international conference on Flexible Query Answering Systems*, pages 343–355. Springer-Verlag, 2006.

[DVMT13] Roberto De Virgilio, Antonio Maccioni, and Riccardo Torlone. A similarity measure for approximate querying over rdf data. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, pages 205–213. ACM, 2013.

[ERW11] Shady Elbassuoni, Maya Ramanath, and Gerhard Weikum. Query relaxation for entity-relationship search. In *Proceedings of the 8th Extended Semantic Web Conference on The Semanic Web: Research and Applications*, pages 62–76. Springer Berlin Heidelberg, 2011.

[ES13a] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, 2nd edition, 2013.

[ES13b] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2nd edition, 2013.

[ESZ07] Jérôme Euzenat, François Scharffe, and Antoine Zimmermann. D2. 2.10: Expressive alignment language and implementation. *Knowledge Web project report, KWEB/2004/D2.2.10/1.0.*, 2007.

[FCPW17] Riccardo Frosini, Andrea Calì, Alexandra Poulovassilis, and Peter T Wood. Flexible query processing for sparql. *Semantic Web*, 8(4):533–563, 2017.

[FF14] Takahisa Fujino and Naoki Fukuta. Utilizing weighted ontology mappings on federated sparql querying. In Wooju Kim, Ying Ding, and Hong-Gee Kim, editors, *Semantic Technology*, Lecture Notes in Computer Science, pages 331–347. Springer International Publishing, 2014.

[GJM09] Hugh Glaser, Afraz Jaffri, and Ian Millard. Managing co-reference on the semantic web. In *WWW2009 Workshop: Linked Data on the Web (LDOW2009)*, April 2009. Event Dates: 20 April 2009.

[GKL01] Zong Woo Geem, Joong Hoon Kim, and GV Loganathan. A new heuristic optimization algorithm: harmony search. *Simulation*, 76(2):60–68, 2001.

[GS11] Olaf Görlitz and Steffen Staab. Splendid: Sparql endpoint federation exploiting void descriptions. In *Proceedings of the Second International Conference on Consuming Linked Data-Volume 782*, pages 13–24. CEUR-WS. org, 2011.

[GTHP13] Pascal Gillet, Cassia Trojahn, Ollivier Haemmerlé, and Camille Pradel. Complex correspondences for query patterns rewriting. In *Proceedings of the 8th International Conference on Ontology Matching-Volume 1111*, pages 49–60. CEUR-WS. org, 2013.

[GTS12] Olaf Görlitz, Matthias Thimm, and Steffen Staab. Splodge: Systematic generation of sparql benchmark queries for linked open data. In *International Semantic Web Conference*, pages 116–132. Springer, 2012.

[Har11] Olaf Hartig. Zero-knowledge query planning for an iterator implementation of link traversal based query execution. In *Proceedings of the 8th extended semantic web conference on The semantic web: research and applications-Volume Part I*, pages 154–169. Springer-Verlag, 2011.

[HB11] 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.

[HBF09] Olaf Hartig, Christian Bizer, and Johann-Christoph Freytag. Executing sparql queries over the web of linked data. In *International Semantic Web Conference*, pages 293–309. Springer, 2009.

[HLZ08] Hai Huang, Chengfei Liu, and Xiaofang Zhou. Computing relaxed answers on rdf databases. In *Web Information Systems Engineering - WISE 2008*, volume 5175 of *Lecture Notes in Computer Science*, pages 163–175. Springer Berlin Heidelberg, 2008.

[HLZ12] Hai Huang, Chengfei Liu, and Xiaofang Zhou. Approximating query answering on rdf databases. *World Wide Web*, 15(1):89–114, 2012.

[HMM13] Bernhard Haslhofer, Flávio Martins, and João Magalhães. Using skos vocabularies for improving web search. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 1253–1258. International World Wide Web Conferences Steering Committee, 2013.

[HMPS12] Aidan Hogan, Marc Mellotte, Gavin Powell, and Dafni Stampouli. Towards fuzzy query-relaxation for rdf. In *ESWC*, volume 7295 of *Lecture Notes in Computer Science*, pages 687–702. Springer, 2012.

[HMZ10] Peter Haase, Tobias Mathäß, and Michael Ziller. An evaluation of approaches to federated query processing over linked data. In *Proceedings of the 6th International Conference on Semantic Systems*, I-SEMANTICS '10, pages 5:1–5:9, New York, NY, USA, 2010. ACM.

[HPS14] Patrick J Hayes and Peter F Patel-Schneider. Rdf 1.1 semantics. w3c recommendation, february 2014. *World Wide Web Consortium. Retrieved from https://www. w3. org/TR/2014/REC-rdf11-mt-20140225*, 2014.

[HPW08] Carlos Hurtado, Alexandra Poulovassilis, and Peter Wood. Query relaxation in rdf. *Journal on Data Semantics X*, pages 31–61, 2008.

[HWM+17] Konrad Höffner, Sebastian Walter, Edgard Marx, Ricardo Usbeck, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. Survey on challenges of question answering in the semantic web. *Semantic Web*, 8(6):895–920, 2017.

[JHY+10] Prateek Jain, Pascal Hitzler, Peter Z Yeh, Kunal Verma, and Amit P Sheth. Linked data is merely more data. In *AAAI Spring Symposium: linked data meets artificial intelligence*, volume 11, 2010.

[KM15] Mayank Kejriwal and Daniel P. Miranker. An unsupervised instance matcher for schema-free {RDF} data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 35, Part 2:102 – 123, 2015. Machine Learning and Data Mining for the Semantic Web (MLDMSW).

[LB11] Jens Lehmann and Lorenz Bühmann. Autosparql: let users query your knowledge base. In *Proceedings of the 8th extended semantic web conference on The semantic web: research and applications-Volume Part I*, pages 63–79. Springer, 2011.

[LFMS12] Vanessa Lopez, Miriam Fernández, Enrico Motta, and Nico Stieler. Poweraqua: Supporting users in querying and exploring the semantic web. *Semantic Web*, 3(3):249–265, 2012.

[LT10] Günter Ladwig and Thanh Tran. Linked data query processing strategies. In *Proceedings of the 9th international semantic web conference on The semantic web-Volume Part I*, pages 453–469. Springer-Verlag, 2010.

[LTLL09] Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. Rimom: A dynamic multistrategy ontology alignment framework. *IEEE Transactions on Knowledge and Data Engineering*, 21(8):1218–1232, 2009.

[MBGC12a] Konstantinos Makris, Nikos Bikakis, Nektarios Gioldasis, and Stavros Christodoulakis. Sparql-rw: transparent query access over mapped rdf data sources. In *Proceedings of the 15th International Conference on Extending Database Technology*, pages 610–613. ACM, 2012.

[MBGC12b] Konstantinos Makris, Nikos Bikakis, Nektarios Gioldasis, and Stavros Christodoulakis. Sparql-rw: transparent query access over mapped rdf data sources. In *Proceedings of the 15th International Conference on Extending Database Technology*, pages 610–613. ACM, 2012.

[MLAN11] Mohamed Morsey, Jens Lehmann, Sören Auer, and Axel-Cyrille Ngonga Ngomo. Dbpedia sparql benchmark–performance assessment with real queries on real data. In *International Semantic Web Conference*, pages 454–469. Springer, 2011.

[MMM+04] Frank Manola, Eric Miller, Brian McBride, et al. Rdf primer. *W3C recommendation*, 10(1-107):6, 2004.

[MRT12] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[OD15] David J Odgers and Michel Dumontier. Mining electronic health records using linked data. *AMIA Summits on Translational Science Proceedings*, 2015:217, 2015.

[PLC+08] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. In *Journal on data semantics X*, pages 133–173. Springer, 2008.

[PSW16] Alexandra Poulovassilis, Petra Selmer, and Peter T Wood. Approximation and relaxation of semantic web path queries. *Web Semantics: Science, Services and Agents on the World Wide Web*, 40:1–21, 2016.

[QL08a] Bastian Quilitz and Ulf Leser. Querying distributed rdf data sources with sparql. In *European Semantic Web Conference*, pages 524–538. Springer, 2008.

[QL08b] Bastian Quilitz and Ulf Leser. Querying distributed rdf data sources with sparql. In *Proceedings of the 5th European semantic web conference on The semantic web: research and applications*, pages 524–538. Springer-Verlag, Springer, 2008.

[Res95] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[RK13] Kuldeep BR Reddy and P Sreenivasa Kumar. Efficient trust-based approximate sparql querying of the web of linked data. In *Uncertainty Reasoning for the Semantic Web II*, pages 315–330. Springer, 2013.

[RLHJ+99] Dave Raggett, Arnaud Le Hors, Ian Jacobs, et al. Html 4.01 specification. *W3C recommendation*, 24, 1999.

[SdM08] Marta Sabou, Mathieu d'Aquin, and Enrico Motta. Scarlet: semantic relation discovery by harvesting online ontologies. In *Proceedings of the 5th European semantic web conference on The semantic web: research and applications*, pages 854–858. Springer-Verlag, 2008.

[SFN11] François Scharffe, Alfio Ferrara, and Andriy Nikolov. Data linking for the semantic web. *International Journal on Semantic Web and Information Systems*, 7(3):46–76, 2011.

[SGH+11a] Michael Schmidt, Olaf Görlitz, Peter Haase, Günter Ladwig, Andreas Schwarte, and Thanh Tran. Fedbench: A benchmark suite for federated semantic data query processing. In *The Semantic Web–ISWC 2011*, pages 585–600. Springer, 2011.

[SGH+11b] Michael Schmidt, Olaf Görlitz, Peter Haase, Günter Ladwig, Andreas Schwarte, and Thanh Tran. Fedbench: a benchmark suite for federated semantic data query processing. In *Proceedings of the 10th ISWC-Volume Part I*, pages 585–600. Springer, 2011.

[SHH+11a] Andreas Schwarte, Peter Haase, Katja Hose, Ralf Schenkel, and Michael Schmidt. Fedx: Optimization techniques for federated query processing on linked data. In *International Semantic Web Conference*, pages 601–616. Springer, 2011.

[SHH+11b] Andreas Schwarte, Peter Haase, Katja Hose, Ralf Schenkel, and Michael Schmidt. Fedx: optimization techniques for federated query processing on linked data. In *Proceedings of the 10th ISWC-Volume Part I*, pages 601–616. Springer, 2011.

[SHLP09] Michael Schmidt, Thomas Hornung, Georg Lausen, and Christoph Pinkel. Sp^ 2bench: a sparql performance benchmark. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, pages 222–233. IEEE, 2009.

[SMNA14] Saeedeh Shekarpour, Edgard Marx, Axel-Cyrille Ngonga Ngomo, and Sören Auer. Sina: Semantic interpretation of user queries for question answering on interlinked data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 0(0):–, 2014.

[SS10] Oshani Seneviratne and Rachel Sealfon. Querymed: An intuitive federated sparql query builder for biomedical rdf data. *MIT report*, 2010.

[SSB+14] Kai Schlegel, Florian Stegmaier, Sebastian Bayerl, Michael Granitzer, and Harald Kosch. Balloon fusion: Sparql rewriting based on unified co-reference information. In *Data Engineering Workshops (ICDEW), 2014 IEEE 30th International Conference on*, pages 254–259. IEEE, 2014.

[Sta18] W3C Standard. VOCABULARIES. `https://www.w3.org/standards/semanticweb/ontology`, 2018. [Online; accessed 07-October-2018].

[SWG13a] W3C SPARQL Working Group. Sparql 1.1 overview., 2013. W3C Recommendation 21 March 2013.

[SWG13b] W3C SPARQL Working Group. Sparql 1.1 query language., 2013. W3C Recommendation 21 March 2013.

[Tae04] Gabriele Taentzer. Agg: A graph transformation environment for modeling and validation of software. In *Applications of Graph Transformations with Industrial Relevance*, pages 446–453. Springer, 2004.

[TB13] Ana I Torre-Bastida. Incremental sparql query processing. In *Extended Semantic Web Conference*, pages 712–716. Springer, 2013.

[TB18a] Ana Isabel Torre Bastida. Sparql query set (pdf format), Feb 2018. Available at `https://figshare.com/articles/SPARQL_Query_Set/5900236/1`.

[TB18b] Ana Isabel Torre Bastida. Sparql query set (txt format), Feb 2018. Available at `https://figshare.com/articles/100SPARQLgold_adequate_source_target_txt/5896627/1`.

[TBBI+13] Ana I Torre-Bastida, Jesús Bermúdez, Arantza Illarramendi, Eduardo Mena, and Marta González. Query rewriting for an incremental search in heterogeneous linked data sources. In *International Conference on Flexible Query Answering Systems*, pages 13–24. Springer, 2013.

[TBBI15] Ana I. Torre-Bastida, Jesús Bermúdez, and Arantza Illarramendi. Query approximation in the case of incompletely aligned datasets. *Actas de las XX Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2015)*, 2015.

[TBBI18a] Ana I Torre-Bastida, Jesús Bermúdez, and Arantza Illarramendi. Estimating query rewriting quality over lod. *Semantic Web*, -(Preprint):1–26, 2018.

[TBBI18b] Ana I. Torre-Bastida, Jesús Bermúdez, and Arantza Illarramendi. A rule-based transducer for querying incompletely aligned datasets. *ACM Trans. Web*, 12(4):23:1–23:40, September 2018.

[TBBMI11] Ana I Torre-Bastida, Jesús Bermúdez, Eduardo Mena, and Arantza Illarramendi. Diseño de un repositorio rdf basado en tecnologías nosql. In *JISBD 2011. XVI Jornadas de Ingeniería del Software y Bases de Datos*, pages –. Springer, 2011.

[TGEM07] Raquel Trillo, Jorge Gracia, Mauricio Espinoza, and Eduardo Mena. Discovering the semantics of user keywords. *J. UCS*, 13(12):1908–1935, 2007.

[TUY11] Yuan Tian, Jürgen Umbrich, and Yong Yu. Enhancing source selection for live queries over linked data via query log mining. In *Proceedings of the 2011 joint international conference on The Semantic Web*, pages 176–191. Springer-Verlag, Springer, 2011.

[WEG87] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[WP94] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

[ZDCC12] Ying Zhang, Pham Minh Duc, Oscar Corcho, and Jean-Paul Calbimonte. Srbench: a streaming rdf/sparql benchmark. In *International Semantic Web Conference*, pages 641–657. Springer, 2012.

[ZGB+17] Ziqi Zhang, Anna Lisa Gentile, Eva Blomqvist, Isabelle Augenstein, and Fabio Ciravegna. An unsupervised data-driven method to discover equivalent relations in large linked datasets. *Semantic Web*, 8(2):197–223, 2017.

[ZZLY02] Jiwei Zhong, Haiping Zhu, Jianming Li, and Yong Yu. Conceptual graph matching for semantic search. In *International Conference on Conceptual Structures*, pages 92–106. Springer, 2002.