



Universidad Católica
San Pablo

Conversión de texto a RDF

XXxxxx

Asesor: Prof. Dr. Irvin Dongo
Co-asesor: Prof. Dr. Regina Ticona-Herrera
Co-asesor: Prof. Dr. Yudith Cardinale

Tesis de titulación presentada a la Escuela Profesional de Ciencia de la Computación como parte de los requisitos para obtener el Título Profesional de Lic. en Ciencia de la Computación.

UCSP- Universidad Católica San Pablo
Abril de 2024

Aquí deberás colocar a quien va dedicada tu tesis por ejemplo: A Dios, por todo lo que me ha dado, a todos los profesores por sus enseñanzas y algunos amigos.

UNIVERSIDAD CATÓLICA SAN PABLO
FACULTAD DE INGENIERÍA Y COMPUTACIÓN
ESCUELA PROFESIONAL DE CIENCIA DE LA
COMPUTACIÓN

Conversión de texto a RDF

Tesis de titulación presentada por el bachiller XXxxxx en el cumplimiento de los requisitos para obtener el título profesional de Licenciado en Ciencia de la Computación.

Arequipa, 25 de abril de 2024

Aprobado por:

Prof. Dr. Hidalgo Buena Gente
PRESIDENTE

Prof. Dr. Manuel Armando Líos
VOCAL

Prof. Dr. Antero A. Gal Oppe
SECRETARIO

Prof. Dr. Casso E. Staria
EXTERNO
Universidad del ABC

Abreviaturas

RDF Resource Description Framework

BDR Base de Datos Relacional

XML Lenguaje de Marcado Extensible

CSV Valores Separados por Coma

RDFs Esquema RDF

IRI Identificadores de Recursos Internacionalizados

TSV Valores Separados por Tabuladores

NLP Procesamiento de Lenguaje Natural

Resumen

Aquí deberás colocar entre 100 y 150 palabras como máximo, el problema que intentas resolver, la justificación y los aportes o soluciones que planteas y qué tanto los haz alcanzado.

Abstract

Here you must enter between 100 and 150 words maximum, the problem you are trying to solve, the justification and the contributions or solutions you are proposing and how much you have achieved them.

Índice general

1. Introducción	2
1.1. Motivación y Contexto	3
1.2. Planteamiento del Problema	3
1.3. Objetivo General	4
1.4. Objetivos Especifico	4
1.5. Organización de la tesis	4
2. Marco Teórico	5
2.1. Información en la Web	5
2.1.1. Tipos de Datos	5
2.1.2. Datos estructurados	5
2.1.3. Datos semi estructurados	5
2.1.4. Datos no estructurados	6
2.2. La Web Semántica	7
2.2.1. Framework de Descripción de Recursos (RDF)	7
2.2.2. RDFs	10
2.2.3. Ontologías	10
2.2.4. SPARQL	11
3. Estado del Arte	12
3.1. Consideraciones Finales	14

4. Propuesta	17
4.1. Pre-procesamiento	18
4.2. Extracción de Conocimiento	19
4.3. Representación de los datos	20
5. Pruebas y Resultados	22
5.1. Conjunto de Datos	22
5.2. Implementación	22
5.2.1. Pre-procesamiento	22
5.2.2. Extracción de conocimiento	23
5.2.3. Representación de datos	24
5.3. Experimentos	24
5.3.1. Pruebas en la fase de preprocesamiento	24
5.3.2. Pruebas para la fase de extracción de conocimiento	26
6. Conclusiones	28
6.1. Trabajos futuros	29
Bibliografía	33

Índice de tablas

3.1. Comparación del estado del arte	15
5.1. Cantidad de nombres de entidades unidos correcta e incorrectamente. . . .	25

Índice de figuras

2.1. Un grafo con dos nodos y un enlace que los conecta	7
2.2. Fragmento de la representación en XML/RDF	8
2.3. Fragmento de la representación en Ntriple	9
2.4. Fragmento de la representación en Turtle	9
2.5. Fragmento de la representación en JsonLD	9
4.1. Arquitectura de la propuesta	17
4.2. Arquitectura de la fase de pre-procesamiento	19
4.3. Arquitectura de la fase de extracción de conocimiento	20
4.4. Arquitectura de la fase de representación de datos	21
5.1. Resultados de StanfordCoreNLP para la oración 1, utilizando Senna en la imagen de la izquierda	24
5.2. Resultados de StanfordCoreNLP para la oración 2, utilizando Senna en la imagen de la izquierda	25
5.3. Resultados de StanfordCoreNLP para la oración 3, utilizando Senna en la imagen de la izquierda	25

Capítulo 1

Introducción

En la actualidad los datos que se publican en Internet aumentan cada día de manera exponencial y además gran parte de esta información carece de estructura alguna, como son texto, video, música, etc. Esto genera que la búsqueda y recuperación de la información sea cada vez más compleja debido a la cantidad y a la redundancia de ella. Esta información muchas veces no está conectada y tampoco tienen descripciones que nos puedan dar un indicio de lo que realmente significan. Esto trae como consecuencia que al momento de realizar consultas en cualquier motor de búsqueda, la información que nos devuelve muchas veces no es exactamente lo que uno desea. Es por ello que se necesita analizar el sentido semántico de las consultas para una respuesta óptima y eficiente. La Web Semántica surge como un esfuerzo conjunto para solucionar estos problemas. Existen diferentes iniciativas que nos permiten convertir datos no estructurados a estructurados, y en particular a datos de tipo Resource Description Framework (RDF).

RDF es un modelo estándar para el intercambio de datos en la Web, está conformado por tripletas (sujeto, Predicado, objeto) $\langle s, p, o \rangle$ esenciales para el análisis semántico [Salas et al., 2011]. Con este esquema simple, RDF permite que los datos estructurados (bases de datos relacionales) y semiestructurados (Lenguaje de Marcado Extensible (XML), Valores Separados por Coma (CSV)) se puedan combinar y sobre todo utilizarse en cualquier aplicación [Group, 2014]. Sin embargo, es posible también transformar datos no estructurados a RDF, especialmente si nos referimos a texto plano, que además es el tipo de datos cuyo volumen es el mayor en la Web. RDF además nos ayuda a encontrar sentido semántico a las palabras, proporcionándonos más ventajas como la posibilidad de describir las propiedades de los documentos y recursos y la interoperabilidad entre distintas aplicaciones sin pérdida del significado de los datos.

Al día de hoy, existen varias investigaciones en la transformación de datos estructurados y semiestructurados a RDF con el objetivo de proporcionar un significado semántico y conectar los datos [Hert et al., 2011][Michel et al., 2014][Breitling, 2009a]. Sin embargo, existe también la necesidad de convertir datos no estructurados a RDF, especialmente el texto plano, representando un reto diferente con respecto a los demás tipos de datos ya que implica un pre-procesamiento sobre el texto para poder extraer ciertas características que nos puedan ayudar al momento de la conversión a RDF. Este pre-procesamiento define en cierta forma la calidad de nuestras tripletas generadas en la conversión.

Este pre-procesamiento representa además una de las limitaciones más grandes para la conversión de texto plano a RDF, usándose herramientas basadas en Procesamiento de Lenguaje Natural (NLP), para tratar de encontrar cierta estructura en los textos. A raíz de esta limitación, los trabajos existentes se enfocan principalmente en la conversión de texto plano a RDF ya que todavía sigue siendo un reto realizar dicha conversión.

En este contexto, este trabajo propone una mejora en el proceso de conversión de texto plano a RDF, realizando un análisis profundo de las limitaciones de las técnicas actuales.

1.1. Motivación y Contexto

En la actualidad, el 80 % de los datos publicados en la Web son no estructurados [Blog, 2018], lo que limita el análisis de los datos para su uso en diversas aplicaciones. Un reporte realizado por Microfocus [Schultz, 2019], muestra la inmensa cantidad de información que tenemos hoy en día en la Web:

- Desde 2013, el número de Tweets por minuto ha aumentado un 58 % a más de 474,000 Tweets por minuto en el 2019.
- Hay más de 4 millones de horas de contenido subidas a Youtube todos los días.
- Existe 4.3 billones de mensajes de Facebook publicados diariamente.
- 26 millones de textos fueron enviados cada día por 27 millones de personas en los Estados Unidos.

Como se observa en el informe presentado por Microfocus, existe demasiada información no estructurada y semi-estructurada en la Web, que va creciendo muy rápidamente. Es por ello la necesidad y la importancia de convertir texto a RDF, para poder realizar un análisis más profundo y certero de la información en diversos campos, además de conectar la información.

1.2. Planteamiento del Problema

Debido a la gran cantidad de información disponible en Internet y sobre todo a la existencia de datos no estructurados, generando información ambigua, existe la necesidad de estructurar esta información para su posterior análisis y uso. En el estado de arte sobre las técnicas de conversión de texto plano a RDF, existen diversas problemas y limitaciones que aún no han sido resueltas y pueden mejorar, como el caso del pre-procesamiento de la información y la generación de tripletas poco coherentes. Una excesiva generación de tripletas tiende a disminuir la utilidad de los datos, ya que genera tripletas redundantes, lo que nos indica lo importante que es el pre-procesamiento.

1.3. Objetivo General

El objetivo de esta investigación es mejorar el proceso de conversión de texto a RDF; es decir, desarrollar un enfoque que permita generar una cantidad correcta de tripletas a partir de texto plano.

1.4. Objetivos Especifico

- Realizar un análisis del estado del arte, para conocer los trabajo existentes sobre conversión de texto a RDF.
- Proponer criterios de comparación que permitan clasificar los trabajos existentes.
- Proponer una mejora en el proceso de conversión de texto plano a RDF.
- Validar la propuesta a través métricas de comparación como la consistencia de tripletas, tiempo de conversión, entre otros.

1.5. Organización de la tesis

Este trabajo está organizado de la siguiente manera:

En el Capítulo 2, se presenta el Marco Teórico, donde se explica los diferentes conceptos acerca de la Web Semántica, RDF y demás tecnologías relacionados con esta investigación.

En el Capítulo 3, se presenta el Estado del Arte en el contexto de transformación de texto a RDF, describiendo brevemente los diferentes enfoques de transformación que existen en la literatura.

En el Capítulo 4, se describe la propuesta para la conversión de texto plano a documentos RDF.

En el Capítulo 5, se realiza una evaluación experimental de la fase de pre-procesamiento utilizando el algoritmo de Senna.

En el Capítulo 6, se describen las conclusiones preliminares de la implementación parcial de la propuesta.

Capítulo 2

Marco Teórico

En este capítulo se presenta los tipos de datos que se encuentran en la Web, estructurados, semi estructurados y no estructurados (sección 2.1). Además, se desarrolla los temas relacionados a la Web Semántica en la sección 2.2 y sus tecnologías como son RDF, serializaciones, RDFs, Ontologías y SPARQL.

2.1. Información en la Web

2.1.1. Tipos de Datos

En el contexto de la Web existen principalmente 3 tipos de datos: (i) datos estructurados, (ii) semiestructurados y (iii) no estructurados. Cada uno de ellos tiene sus propias características que son descritas en las secciones siguientes, centrándonos principalmente en los datos no estructurados.

2.1.2. Datos estructurados

Los datos estructurados son aquellos valores de datos que se ajustan a un esquema o tipo común bien especificado. Un ejemplo común son las bases de datos relacionales. Estos datos por tener una estructura definida, son más accesibles para la conversión a RDF, existiendo varios trabajos que realizan este proceso, un claro ejemplo de esto es la conversión de bases de datos relacionales a RDF algunas investigaciones que se desarrollaron con este propósito se muestran en [Hert et al., 2011, Michel et al., 2014] los cuales utilizan diferentes enfoque para realizar la conversión.

2.1.3. Datos semi estructurados

En esencia los datos semiestructurados no son datos en bruto; sin embargo, tampoco son datos que tengan una estructura definida. Existen algunas características principales

de los datos semiestructurados, entre ellos está la estructura irregular e implícita (por que no indica lo que exactamente significa). Lógicamente su estructura es parcial por que pueden estar generada a partir de etiquetas.

Entre los datos semiestructurados tenemos los documentos CSV, XML, JSON, bases de datos NoSQL, HTML, entre otros. Si bien el proceso de conversión es diferente al de datos estructurados, al contar con una estructura, aunque no está totalmente definida, facilita el proceso de conversión. Existen varios enfoques centrados en este tipo de datos, especialmente la conversión de datos XML a RDF, esto debido a que en la actualidad XML es un estandar para el intercambio de informacion a traves de la web bien establecido y bastante utilizado por que mucha información se encuentra en este formato, en los siguientes trabajos se desarrollaron enfoques que realizaron esta conversion [Huang et al., 2015, Deursen et al., 2008a, Breitling, 2009b]. De la misma manera los archivos con valores separados por comas (CSV) son bastante utilizado para publicar datos en la actualidad, y muchos organismos lo utilizan con grandes e importantes cantidades de datos, es por ello que se busca transformas esto datos a RDF, algunos trabajos que han propuesto enfoques para este proposito son presentados por [Ermilov et al., 2013, Mahmud et al., 2018].

2.1.4. Datos no estructurados

Los datos no estructurados son aquellos que carecen de algún tipo de estructura, siendo principalmente los textos planos que se encuentran en revistas, diarios, libros, obras, entre otros. Estos son los datos con los que más están acostumbradas las personas, lo que significa además que es la forma más común de almacenamiento de datos. De hecho, en el Internet también se puede apreciar este suceso, ya que existe un 80 % de datos en la forma de texto plano, es decir datos no estructurados. Esto es una desventaja al momento de querer analizar los datos o recuperar información, ya que no existe una manera de analizarlos sin realizar antes un pre-procesamiento. Todos los trabajos que buscan una forma de transformar datos no estructurados a RDF destacan esto y usan algún enfoque para realizar un análisis y pre-procesamiento, entre ellos hacen el uso de Procesamiento de Lenguaje Natural (PLN) algunos con Inteligencia Artificial o Machine Learning o cualquier otro método que permita analizar el texto y tratar de reconocer patrones o la semántica de este, una vez realizado esto es posible convertir a tripletas RDF, algunos autores han propuesto enfoques para convertir texto a RDF, entre ellos algunos están [Exner and Nagues, 2012, Augenstein et al., 2012, Hassanzadeh et al., 2013, Draicchio et al., 2013]

A raíz de esta necesidad existen varios enfoques que nos permiten analizar e interpretar estos datos, técnicas tales como, minería de datos, procesamiento de lenguaje natural (NLP por sus siglas en inglés) o el análisis de texto en sí [Holzinger et al., 2013].

Existen varios tipos de datos no estructurados tales como correos electrónicos, archivos de procesador de texto, archivos PDF, imágenes digitales, vídeo, audio, publicaciones en medios sociales y muchas más. Sin embargo, nuestro enfoque se centra en datos de texto plano, ya que requiere de un mayor procesamiento para realizar la conversión a RDF. Además se puede obtener una mayor aplicabilidad si se analiza toda esa información, al

momento por ejemplo de devolver respuestas a consultas en la Web.

2.2. La Web Semántica

Tim Berners-Lee en su investigación [Berners-Lee and Hendler, 2001], presentó las bases de la Web Semántica, a raíz de que los datos crecían en grandes volúmenes en la Web; sin embargo, gran parte de ellos carecían y carecen hasta la actualidad de estructura alguna, lo que limita totalmente el procesamiento de estos datos. Con este problema se piensa en una forma de agregar semántica a los datos de tal manera que se pueda explotar los datos y facilitar la interoperabilidad, ya que esta gran cantidad de datos e información podría traer grandes oportunidades y beneficios. Evidentemente todo esto exige un gran procesamiento en las aplicaciones, más aún si los datos carecen completamente de estructura como es el texto plano, es por ello que se empieza a utilizar datos más estructurados como es el caso de RDF para darles un formato a los datos publicados en la Web, pudiendo localizarlos y procesarlos [Laufer, 2015].

2.2.1. Framework de Descripción de Recursos (RDF)

RDF es considerado como la piedra angular sobre la que se construye la estructura semántica de la Web. RDF es definido por la W3C como un modelo estándar para el intercambio y representación de datos en la Web [Group, 2014]. Permite identificar y definir recursos; un recurso puede ser cualquier cosa en la Web como por ejemplo, una empresa, persona, páginas, sentimientos, colores, entre otros [Laufer, 2015]. RDF está conformada por tripletas, $\langle \text{sujeito}, \text{predicado}, \text{objeto} \rangle$ y cada una de estas partes pueden ser, IRIs, nodos en blanco o literales con tipos de datos [Graham Klyne and McBride, 2014]. Esta tripleta puede representarse como un grafo con el sujeto y objeto como nodos y el predicado como enlace. En la figura 2.1 presentamos un ejemplo RDF.



Figura 2.1: Un grafo con dos nodos y un enlace que los conecta

Estructura de una tripleta RDF

Como se ha mencionado anteriormente, una tripleta está formada por 3 partes que pueden ser:

- **IRI.-** Identificadores de Recursos Internacionalizados (IRI) es una secuencia de caracteres del juego de caracteres universal (Unicode / ISO 10646). Es posibles realizar una asignación de IRI a URI, lo que significa que se pueden usar IRI en lugar de URI para identificar recursos en la Web [Duerst, 2004].

- **Literal.-** Son usados para valores como cadenas, números y fechas, consta de 2 o 3 partes, que son: una forma léxica, es decir una cadena UNICODE, un IRI de tipo de datos y si este es string, una etiqueta de idioma [Graham Klyne, 2014].
- **Nodo en blanco.-** También llamado *bnode*, es un nodo en un grafo RDF que representa un recurso pero no se le ha asignado una IRI o un literal [Patrick J. Hayes, 2014].

Para el caso de un sujeto: este puede ser un IRI, o un nodo en blanco, en el caso de un predicado: puede ser únicamente una IRI, mientras que un objeto puede ser: un IRI, literal o nodo en blanco.

Serialización de RDF

Existen varias maneras de representar datos RDF, para ser utilizados en alguna aplicación. Para explicar esto de mejor manera utilizaremos un pequeño fragmento de una oración en FRED (Un enfoque desarrollado por [Draicchio et al., 2013] para la conversión de texto a RDF), el objetivo es ver la salida serializado en distintos formatos:

Nuestra oración de ejemplo es: *"Miles Davis was an american jazz musician."*

Esta oración convertido a RDF se puede serializar en los siguientes formatos:

- **RDF/XML.-** Básicamente es una sintaxis XML pero enfocada para tripletas RDF [Gandon and Schreiber, 2014].

```
<rdf:Description rdf:about="http://www.ontologydesignpatterns.org/ont/fred/domain.owl#offset_0_11_Miles_Davis">
  <j.1:denotes rdf:resource="http://www.ontologydesignpatterns.org/ont/fred/domain.owl#Miles_davis"/>
  <j.6:pennpos rdf:resource="http://www.ontologydesignpatterns.org/ont/fred/pos.owl#NNP"/>
  <j.2:begins rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">0</j.2:begins>
  <rdf:type rdf:resource="http://www.essepuntato.it/2008/12/earmark#PointerRange"/>
  <j.2:ends rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">11</j.2:ends>
  <j.2:refersTo rdf:resource="http://www.ontologydesignpatterns.org/ont/fred/domain.owl#docuverse"/>
  <j.1:hasInterpretant rdf:resource="http://www.ontologydesignpatterns.org/ont/fred/domain.owl#Jazz"/>
  <j.1:hasInterpretant rdf:resource="http://www.ontologydesignpatterns.org/ont/fred/domain.owl#Musician"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Miles_Davis</rdfs:label>
</rdf:Description>
```

Figura 2.2: Fragmento de la representación en XML/RDF

- **N-TRIPLE.-** Es un formato de texto plano basado en líneas para representar un grafo RDF [Carothers and Seaborne, 2014].
- **TURTLE.-** Es una sintaxis textual para RDF que logra que las tripletas sean convertidas a un formato de texto natural y compacta, además permite dar abreviaturas a ciertos patrones o a tipos de datos que podrían existir en una tripleta RDF [David Beckett and Machina, 2014].
- **JSON-LD.-** Es un formato basado en bu JSON, pero enfocado para datos vinculados esto a su vez permite también que sea utilizado como una representación de tripletas RDF [Manu Sporny and Lindström, 2014].

```

<http://www.ontologydesignpatterns.org/ont/fred/domain.owl#offset_28_32_jazz>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.essepuntato.it/2008/12/earmark#PointerRange>
<http://www.ontologydesignpatterns.org/ont/fred/domain.owl#offset_28_32_jazz>
<http://www.w3.org/2000/01/rdf-schema#label>
"jazz"^^<http://www.w3.org/2001/XMLSchema#string>
<http://www.ontologydesignpatterns.org/ont/fred/domain.owl#offset_28_32_jazz>
<http://ontologydesignpatterns.org/cp/owl/semiotics.owl#denotes>
<http://www.ontologydesignpatterns.org/ont/fred/domain.owl#Miles_davis>
<http://www.ontologydesignpatterns.org/ont/fred/domain.owl#offset_28_32_jazz>
<http://ontologydesignpatterns.org/cp/owl/semiotics.owl#denotes>
<http://www.ontologydesignpatterns.org/ont/fred/domain.owl#jazz_1>

```

Figura 2.3: Fragmento de la representación en Ntriple

```

<http://www.w3.org/2000/01/rdf-schema#label>
    "jazz"^^<http://www.w3.org/2001/XMLSchema#string> ;
<http://ontologydesignpatterns.org/cp/owl/semiotics.owl#denotes>
    <http://www.ontologydesignpatterns.org/ont/fred/domain.owl#Miles_davis>
<http://ontologydesignpatterns.org/cp/owl/semiotics.owl#hasInterpretant>
    <http://www.ontologydesignpatterns.org/ont/fred/domain.owl#Jazz> ;
<http://www.essepuntato.it/2008/12/earmark#begins>
    "28"^^<http://www.w3.org/2001/XMLSchema#nonNegativeInteger> ;
<http://www.essepuntato.it/2008/12/earmark#ends>
    "32"^^<http://www.w3.org/2001/XMLSchema#nonNegativeInteger> ;
<http://www.essepuntato.it/2008/12/earmark#refersTo>
    <http://www.ontologydesignpatterns.org/ont/fred/domain.owl#docuverse> ;
<http://www.ontologydesignpatterns.org/ont/fred/pos.owl#pennpos>
    <http://www.ontologydesignpatterns.org/ont/fred/pos.owl#NN> .

```

Figura 2.4: Fragmento de la representación en Turtle

```

[{"@id":"http://dbpedia.org/ontology/Place",
 {"@id":"http://dbpedia.org/resource/Jazz",
 {"@id":"http://dbpedia.org/resource/Miles_Davis",
 "@type":["http://schema.org/MusicGroup","http://schema.org/Person"]},
 {"@id":"http://dbpedia.org/resource/Musician",
 {"@id":"http://ontologydesignpatterns.org/cp/owl/semiotics.owl#denotes",
 "@type":["http://www.w3.org/2002/07/owl#ObjectProperty"]},
 {"@id":"http://ontologydesignpatterns.org/cp/owl/semiotics.owl#hasInterpretant",
 "@type":["http://www.w3.org/2002/07/owl#ObjectProperty"]},
 {"@id":"http://schema.org/MusicGroup"}, {"@id":"http://schema.org/Person"}]}]

```

Figura 2.5: Fragmento de la representación en JsonLD

En las figuras 2.2, 2.3, 2.4 y 2.5 se han visto distintos tipos de serialización para RDF, y un conjunto de tripletas representa un documento RDF, donde las tripletas contenidas no se repiten, y además no tienen un orden definido [Bizer et al., 2018].

2.2.2. RDFs

El esquema RDF o Esquema RDF (RDFs) por sus siglas en inglés, proporciona un vocabulario de modelado para datos RDF. Además explica se define como se relacionan los nodos de un grafo RDF [W3, 2016]. A continuación, se presenta las definiciones que realiza un vocabulario:

- Define superclases y subclases.
- Indica que conexiones están permitidas.
- Clases de documentos y propiedades, con etiquetas y comentarios.
- Indica cómo buscar otros segmentos del grafo
- Hablar sobre contenedores y membresía en el resumen

2.2.3. Ontologías

Si bien RDF nos ayuda a estructurar datos, las ontologías se encargan de hacer una semántica para construirlos. Una ontología es una especificación de una conceptualización, es decir, un marco común y de consenso no sólo para almacenar la información, sino también para poder buscarla y recuperarla [Lapiente, 2013]. otra definición de ontología es un vocabulario acerca de un dominio específico: términos + relaciones + reglas de combinación para extender el vocabulario. Gruber [GRUBER., 1993] menciona que la ontología está conformada por:

- Conceptos: son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.
- Relaciones: representan la interacción y enlace entre los conceptos de un dominio. Suelen formar la taxonomía del dominio. Por ejemplo: subclase-de, parte-de, parte-exhaustiva-de, conectado-a, etc.
- Funciones: son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo, pueden aparecer funciones como: asignar-fecha, categorizar-clase, etc.
- Instancias: se utilizan para representar objetos determinados de un concepto.
- reglas de restricción o axiomas: son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología.

2.2.4. SPARQL

Es un conjunto de especificaciones que proporcionan lenguajes y protocolos para consultar y manipular el contenido RDF en la Web o en una base de datos RDF. Una vez realizado una consulta las respuestas convenientemente son representadas en forma de tabla, y pueden ser presentados además en formato XML, JSON, CSV o Valores Separados por Tabuladores (TSV) [Group, 2013].

Para realizar una consulta primero hay que tener en cuenta que SPARQL tiene un conjunto de patrones triples llamados patrón de gráfico básico, los cuales son similares a RDF pero con la diferencia de que tanto el sujeto, predicado y objeto puede ser una variable. A continuación presentamos una ejemplo de una consulta de SPARQL:

RDF:

<http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> "Tutorial de SPARQL"

Consulta:

SELECT ?title

WHERE

<http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> ?title

Resultado:

title: Tutorial de SPARQL

Como podemos darnos cuenta, las consultas son similares a las utilizadas en las bases de datos, y los resultados de estas consultas son mucho mas precisas al momento de buscar información.

En la siguiente sección se describen los trabajos existentes sobre la conversión de datos a RDF, enfocándonos en los tipos de datos no estructurados.

Capítulo 3

Estado del Arte

El objetivo de la Web Semántica es encontrar sentido semántico a los datos de la Web. La Web Semántica coexiste con los sistemas de bases de datos relacionales (Base de Datos Relacional (BDR)) y su esquema que se centran principalmente en el nivel de estructura [Thu, , Hert et al., 2011, Salas et al., 2011]. El mapeo de las bases de datos relacionales a RDF es de gran interés desde el inicio de la Web Semántica como lo menciona T. B. Lee et al. [Berners-Lee, 2009]. A lo largo de los años se ha buscado integrar los datos estructurados con la Web Semántica, es por ello que se han propuesto muchos enfoques (automático o manual) para lograr la integración [Hazber et al., 2016]. Matthias Hert et al. nos muestra algunos de estos enfoques en su investigación donde nos presentan una comparación de distintos lenguajes de mapeo de BDR a RDF como por ejemplo eD2R un lenguaje para asignaciones de BDR a RDF en base a consultas SQL y funciones de transformación que se pueden aplicar a los valores extraídos, otro lenguaje extensible y totalmente declarativo es R2O para realizar asignaciones de una BDR a RDF, también se describe D2RQ otro lenguaje para realizar el mapeo de BDR a RDF con el objetivo de llevar estos datos a la web semántica y que sean accesibles por medio de SPARQL [Hert et al., 2011].

Otro trabajo desarrollado por Franck Michel et al. [Michel et al., 2014] realiza una investigación de distintas técnicas de transformación de BDR a RDF y muestra una clasificación de técnicas en base a los que utilizan R2RML (Lenguaje de Mapeo BDR2RDF) y los que no utilizan R2RML. Algunos de estos son Datalift que nos presenta un conjunto de herramientas bien integradas que nos facilitan el proceso de transformación de distintos formatos (BDR, CSV, XML). También METAmorphoses un proceso de transformación de BDR en RDF mediante el uso de ontologías ya existentes, otra técnica revisada es R2O/ODEMapster un lenguaje declarativo basado en XML, utiliza mapeo basado en semántica de dominio entre ontologías existentes y bases de datos relacionales, su predecesor es D2RQ para casos donde la similaridad entre la BDR y ontología es bajo.

Stadler Claus [Stadler et al., 2015] nos presentan un lenguaje de mapeo análogo a R2RML llamado Sparqlification Mapping Language (SML), que proporciona una forma intuitiva de declarar mapeos basados en consultas SQL, VIEWS y SPARQL, lo que nos ayuda a tener mejor acceso con SPARQL a bases de datos relacionales.

Hazber Mohamed presenta un enfoque para generar automáticamente documentos

de mapeo R2RML, necesarios para la transformación de BDR a RDF, a partir del esquema BDR [Hazber et al., 2016]. Según R2RML, presentado por W3C, se tiene que realizar una especificación de asignaciones para luego realizar el mapeo de BDR a RDF; sin embargo, esto no es una tarea fácil, en la investigación realizada por Vania V. [Vidal et al., 2014] proponen una estrategia para simplificar la especificación de las asignaciones R2RML a BDR de manera automática usando aserciones de correspondencia.

Por otro lado tenemos los datos semiestructurados, en los que también han sido desarrollado algunas técnicas para la conversión de XML a RDF [Huang et al., 2015, Deursen et al., 2008a]. Algunos se especializan en grandes cantidades de datos usando XPath y se centran principalmente en asignaciones que comprenden plantillas triples RDF que emplean expresiones XPath simples. [Huang et al., 2015]. Breitling F. propone un conjunto de notaciones para asignar el esquema XML al esquema RDF la principal ventaja de este enfoque es garantizar la integridad de la estructura y proporcionar más significado para el documento XML original mientras los transforma automáticamente en RDF [Breitling, 2009b]. Otro enfoque realizado por Deursen V. propone un enfoque genérico para la transformación de datos XML en instancias RDF de una manera dependiente de la ontología, por medio de un documento de mapeo [Deursen et al., 2008b]. Existe también investigación para la conversión de datos en formato CSV a RDF, por ejemplo Ermilov I, presenta un framework para realizar la conversión de CSV a RDF, para ello primero se realiza una conversión automática, y luego puede ser revisado por usuarios humanos [Ermilov et al., 2013]. Mahmud S. presenta otro enfoque de conversión de CSV a RDF especialmente escalable para grandes cantidades de datos [Mahmud et al., 2018].

Por último, están los datos no estructurados, donde se proponen técnicas desarrolladas en las investigaciones presentadas por Exner P. [Exner and Nugues, 2012] donde nos presenta un sistema de extremo a extremo que extrae automáticamente triples RDF que describen las relaciones y propiedades de la entidad del texto no estructurado, este sistema incluye un analizador semántico y un solucionador de coreferencia para agrupar las acciones y propiedades de la entidad descritas en diferentes oraciones y las convertimos en tripletas de entidad. Paso S. nos presenta LODifier [Augenstein et al., 2012] un enfoque que combina un análisis semántico profundo con reconocimiento de entidades con nombre, desambiguación de sentido de palabras y vocabularios controlados de la Web Semántica para extraer entidades con nombre y relaciones entre ellas del texto y convertirlas en una representación RDF. Hassanzadeh K. presenta un sistema que implementa el enfoque es capaz de identificar la estructura gramatical de una oración de entrada y analizar su semántica para generar triples RDF significativos [Hassanzadeh et al., 2013]. FRED es otro sistema muy famoso en el estado del arte para la transformación de texto a RDF, combina la teoría de la representación del discurso (DRT), la semántica de marcos lingüísticos y los patrones de diseño de ontologías (ODP). La herramienta además se basa en Boxer, que implementa un analizador profundo compatible con DRT. La salida lógica de Boxer enriquecida con datos semánticos de cuadros Verbnet o Framenet se transforma en RDF/OWL [Draicchio et al., 2013, Gangemi et al., 2013].

Investigaciones más recientes como la presentada por Martínez J. nos presenta un enfoque para la extracción y representación de declaraciones RDF del texto. Su objetivo es proporcionar una arquitectura que reciba oraciones y devuelva triples con elementos vinculados a recursos y vocabularios de la Web Semántica [Martínez-Rodríguez et al., 2019a]. Un trabajo enfocado en el idioma Árabe es presentado por Zakria G. el sistema pro-

puesto incluye un analizador sintáctico que extrae tripletas del texto árabe preprocesado. Además, el reconocimiento de entidad de nombre se usa para extraer entidades que se mapearon con DBpedia para obtener URI. Finalmente, se genera la representación RDF correspondiente que captura la semántica del texto árabe [Zak, 2019].

En la Tabla 3 se presenta una comparación entre los distintos enfoques del estado del arte que convierten texto a RDF. Se observa que los enfoques se centran en la conversión de documentos en el idioma Inglés, aunque existe una propuesta en el idioma Árabe [Zak, 2019]. Por otro lado Es bastante utilizado y con razón el Procesamiento de Lenguaje Natural, para ellos los diferentes enfoque utilizan técnicas de machine learning, reconocimiento de entidades nombradas y representación teórica del discurso con las técnicas mas utilizadas en los enfoques. Añadimos también una columna donde mencionamos las distintas bases de datos que los enfoques utilizaron en sus pruebas. Todos los enfoques que se encuentran en la Tabla 3 están ordenados de acuerdo al años de su publicación siendo el primero el mas actualizado.

Existen trabajos enfocados específicamente en extraer conocimiento de texto plano; sin embargo, no le dan una estructura, como convertirlo a RDF, por ejemplo, Hazem S. presenta un enfoque para una serie de reglas de reescritura basadas en corpus para la captura de conocimiento posterior. Además se describe un conjunto de datos novedoso que alinea una muestra representativa de oraciones de Wikipedia en inglés simplificado legible por humanos y semánticamente interpretable por una máquina [Abdelaal, 2019]. Otro trabajo realizado por Verma S. donde los algoritmos de aprendizaje automático, es decir, SVM, KNN, regresión logística, regresión lineal y árbol de decisión, se aplican a dichos datos y luego se convierten en una forma estructurada que ayuda en la predicción futura y compara la precisión de cada algoritmo [Verma et al., 2020].

3.1. Consideraciones Finales

A partir del estado del arte revisado, podemos concluir que existen técnicas de conversión de distintos tipos de datos en la Web a RDF, estructurados, semi-estructurados y datos sin estructura alguna. En este último caso, nos enfocamos en los textos planos y se puede ver que todas las técnicas anteriormente desarrolladas incluyen una etapa de pre-procesamiento del texto, considerando este método como una parte fundamental para obtener una conversión consistente. Además todos los enfoques de conversión de texto a RDF mantienen una estructura que consiste en una etapa de pre-procesamiento donde se prepara el texto para ser analizado, en la etapa de extracción de conocimiento se utiliza enfoques para el análisis profundo del texto y la ultima etapa de representación nos permite serializar las tripletas en distintos formatos. Estas 3 fases siempre son utilizados aunque algunas veces pueden tener otras denominaciones.

En cuanto a las métricas de los diferentes enfoque revisados, podemos observar varias métricas para la evaluación de la calidad de tripletas convertidas de texto a RDF; entre ellas las más comunes son: precisión, *recall* y cobertura. Por otro lado, los enfoque como parte de su validación, utilizan una métrica que consiste en una evaluación por parte de un experto; analizando básicamente el resultado a través de criterios propios como parte del conocimiento de un dominio específico; en otras palabras, el experto decide si las tripletas

Tabla 3.1: Comparación del estado del arte

Investigación	Idioma		Técnica			Criterio BD	
	Inglés	Árabe	NLP	Machine Learning	NER	(DRT)	BD
Extraction of RDF Statements from Text	X				X		IT news, LonelyPlanet, BBC news
Knowledge Extraction from Simplified Natural Language Text	X		X				
Entity Extraction: From Unstructured Text to DBpedia RDF triples.	X				X		randomly articles
LODifier: Generating Linked Data from Unstructured Text	X				X	X	TDT-2 benchmark
Semantic Representation Extraction from Unstructured Arabic Text		X			X		60 different articles
T2r: System for converting textual documents into rdf triples	X		X		X		ProMED report
A machine reader for the semantic web	X					X	
Real-time RDF extraction from unstructured data streams	X			X			RdfLiveNews
An Unstructured to Structured Data Conversion using Machine Learning Algorithm in Internet of Things (IoT)	X			X			

generadas son correctas o no.

Capítulo 4

Propuesta

En este capítulo se describe la propuesta, así como la implementación de las diferentes etapas. Para esta investigación se ha tomado en cuenta como base la propuesta realizada por L. Martinez et al. [Martinez-Rodriguez et al., 2019b], donde propone un enfoque basado en 3 aspectos: (i) Pre-procesamiento, (ii) Extracción de Conocimiento y (iii) Representación de datos. Estos aspectos son ampliamente utilizados en la literatura, ya que dividen los diferentes contextos de aplicación en la preparación de los datos, el procesamiento y finalmente en la forma de representación. A continuación en la Figura 4.1 se muestra la arquitectura propuesta.

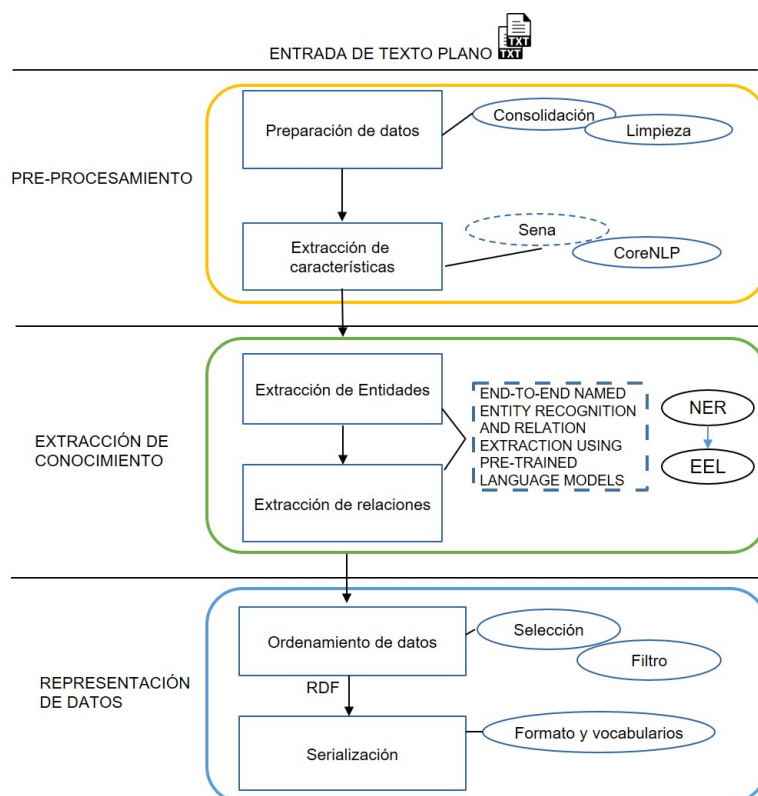


Figura 4.1: Arquitectura de la propuesta

La arquitectura propuesta consta de diferentes etapas, las cuales tienen funciones

específicas y la salida de una es la entrada para la siguiente fase. A continuación se describe fase por fase la funcionalidad del enfoque propuesto.

4.1. Pre-procesamiento

El pre-procesamiento realiza una limpieza del texto plano que ingresa al algoritmo, es decir se eliminan por ejemplo *stop words*. Además, se escriben de forma extendida las palabras que pueden estar en su forma abreviada (ej., acrónimos). Luego se realizan otros procesos como la tokenización de palabras, para que el texto plano tenga mejor formato para el momento de extraer el conocimiento.

Por otro lado, es importante mencionar, en el caso de que el texto de entrada no sea texto plano, sino podría estar en alguno otro formato como XML por ejemplo, esta etapa realizará la respectiva conversión a texto plano. Evidentemente esto no es necesario si se trabaja desde un inicio con el formato deseado.

Por último algo fundamental en esta etapa, es la identificación de nombres, ya sea de ciudades, personas, empresas, organizaciones, etc. Esto con el objetivo de unir los nombres en caso de que estén formados por 2 o más palabras.

A continuación se presenta el algoritmo utilizado para unir los nombre de entidades formados por 2 o más palabras:

Algorithm 1: Algoritmo para unir nombre de entidades

```

Result: Texto con nombres de entidades unidos
O -> Oración;
tokens -> tokenizacionSenna(O);
Of -> Oración final;
for  $i:=0$  to  $len(tokens)$  do
    if  $tokens[i] == name$  and  $tokens[i+1] == name$  then
         $tokens[i] = tokens[i] + tokens[i+1]$ ;
        borrar la posición  $i+1$ ;
    else
end
for  $i:=0$  to  $len(tokens)$  do
     $Of = Of + tokens[i]$ ;
end

```

Como se puede observa en el algoritmo presentado anteriormente, tenemos como entrada una oración cualquiera de nuestro conjunto de datos, esta oración inicial es tokenizado y analizado con la herramienta Senna que nos devuelve cada palabra analizada, donde además reconoce si una palabra es nombre u otro componente de una oración. Una vez obtenido cada palabra si este es un nombre y el que lo sigue también, entonces unimos ambas palabras eliminando el espacio entre ellas, esto se realiza hasta terminar con el conjunto de palabras. Para finalizar se tiene que devolver una Oración con los nombre unidos, para ello simplemente recorremos toda la lista de tokens y los unimos en una sola cadena que devolvemos para continuar con los siguientes procedimientos.

En la Fig. 4.2 se muestra el flujo de la fase de preprocesamiento, en el que se realizan las siguientes tareas:

- Consolidación de textos, en que simplemente unimos textos si se encuentran fragmentados, el objetivo es que estén en un solo texto.
- En caso de existir caracteres desconocidos, estos son eliminados para no generar tripletas incoherentes.
- Lo que continua es reconocer los nombres de cualquier tipo en el texto para que estos sean unidos, con el objetivo de evitar redundancia.
- Lo último de esta fase de preprocesamiento es realizar un análisis más completo del texto plano y limpio, para ello tokenizamos las palabras, en caso de existir abreviaturas o acrónimos estos deben ser extendidos a su nombre completo.

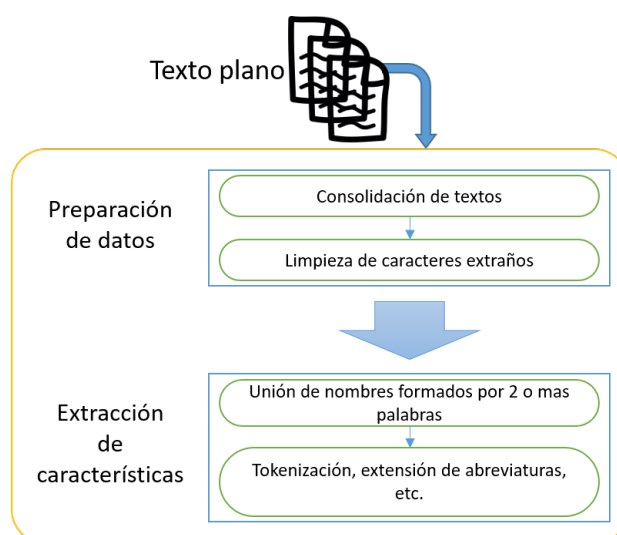


Figura 4.2: Arquitectura de la fase de pre-procesamiento

4.2. Extracción de Conocimiento

Esta es la etapa más importante de los 3 aspectos en nuestro enfoque, ya que esta fase determina la calidad al momento de generar tripletas RDF. En esta etapa se analiza en texto en busca de entidades que pueden estar en forma de sustantivos, como puede ser nombres, lugares, etc. Ya que estos pueden describirse, estas entidades son los recursos en la Web Semántica por ello la importancia de detectarlos ya que pueden formar tripletas.

Es así que la importancia de esta fase es justamente el reconocimiento y extracción de entidades nombradas. Éste es el paso fundamental para la extracción de conocimiento de un texto, y por lo tanto también para la generación de tripletas RDF. Es por ello que en el estado del arte, esta parte ha sido ampliamente estudiada y experimentada para obtener mejores resultados. Además, en el estado del arte se recomienda utilizar un enfoque que realice tanto el reconocimiento y extracción de entidades nombradas como la extracción

de relaciones entre estas entidades, ya que esto genera mejor rendimiento del algoritmo utilizado.

En la Fig. 4.3 detallamos en un diagrama la entrada correspondiente que seria los datos preprocesados de la fase anterior y además se muestra los distintos procedimientos necesarios en esta fase: Lo primero es realizar el reconocimiento de entidades nombradas (NER por sus siglas en ingles). Este tiene como objetivo encontraras todas la entidades en un texto, nos referimos como entidades a sustantivos que pueden ser personas, animales, organización, empresas, etc que pueden ser parte de un texto. Una vez obtenido estas entidades lo que sigue es vincular estas entidades con recursos de la web, es decir son un enlace de la web que puede tener información de la entidades. Para finalizar esta fase del enfoque tenemos que almacenar de algún modo el conjunto de las entidades con sus vínculos correspondientes, ya que son parte para la formación de las tripletas RDF.

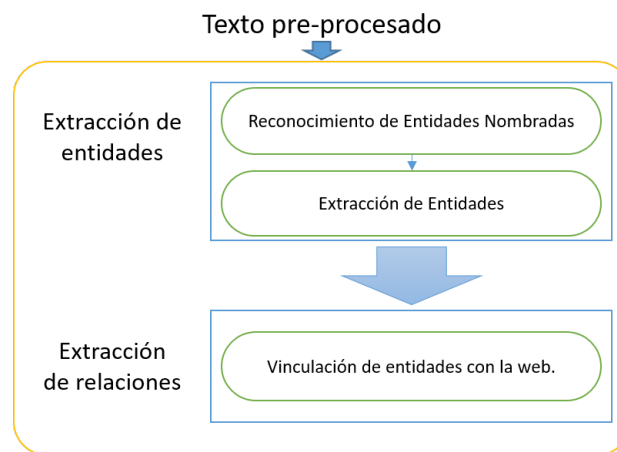


Figura 4.3: Arquitectura de la fase de extracción de conocimiento

4.3. Representación de los datos

En esta última etapa se realizan 2 partes importantes para la transformación. La primera es el proceso de determinar con el conjunto de entidades que tenemos, cuales son sujetos o cuales son objetos en nuestra tripleta final RDF. La segunda es realizar la serialización de los datos, es decir, con las entidades y las relaciones ya obtenidas, buscar una representación de estos datos en algún formato RDF, entre ellos pueden ser, RDF/XML, JSON-LD, N-TRIPLE, etc.

En la Fig. 4.4 se muestra un flujo de la ultima fase de nuestro enfoque en la que describimos los procedimientos y los resultados finales que vamos a obtener en nuestra propuesta. El primer paso en esta fase es ordenar todas las entidades del texto, ya que algunos pueden ser sujetos y otros pueden ser objetos, esto es importante para las tripletas finales, ya que recordando, están formados por <sujetos-predicado-objeto>. Seguidamente diseñamos las tripletas finales, es decir las herramienta que estemos utilizando nos van a pedir una forma especifica de crear un tripleta RDF, es en esta parte donde todas nuestras tripletas lo estructuramos la librería para finalmente serializarlo en cualquier formato como puede ser, RDF/XML, Turtle, N3, JsonLD, etc.

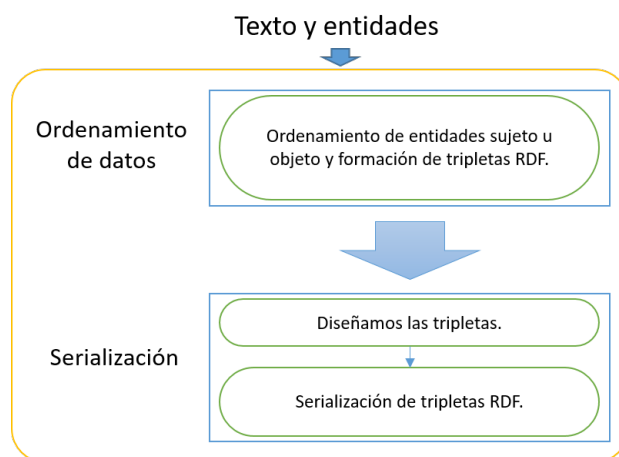


Figura 4.4: Arquitectura de la fase de representación de datos

En el siguiente capítulo se describen las pruebas y resultados de la propuesta.

Capítulo 5

Pruebas y Resultados

En esta sección se describe el proceso de los diferentes experimentos realizados, así como los resultados obtenidos. En estas pruebas se pretende experimentar y validar sobre todo el uso de Senna para la detección y unión de nombres de entidades que estén formadas por 2 o palabras. Esto es comparado con el pre-procesamiento del trabajo más actual [Martinez-Rodriguez et al., 2019a] desarrollado en transformación de texto a RDF.

5.1. Conjunto de Datos

Para el conjunto de datos de los experimentos, se utiliza noticias relacionadas a Tecnologías de la Información (TI). Para ello utilizaremos diferentes fragmentos de noticias de la pagina ComputerWeekly¹. El objetivo de seleccionar fragmentos de textos de noticias, es justamente para comprobar la eficacia de Senna y nuestro algoritmo para detectar nombres conformadas por 2 o mas palabras. Cabe mencionar además que el trabajo utilizado para la comparación, utiliza este mismo conjunto de datos como entrada.

5.2. Implementación

Las herramientas utilizadas para la implementación de la propuesta, son explicadas en esta sección. Para ellos se describe las principales características, de los tres aspectos ya mencionados anteriormente de nuestra arquitectura.

5.2.1. Pre-procesamiento

Es la primera parte de nuestra propuesta, por ello es importante realizar una buena limpieza del texto para que las siguiente fases se realicen de manera óptima. En nuestra arquitectura, preferimos utilizar, Senna y Stanford Core NLP para esta fase. Senna es una

¹<https://www.computerweekly.com/>

herramienta para la detección de nombres ya sea de organizaciones, personas, países, etc. El objetivo principal para el uso de esta herramienta es unir aquellos nombres que estén formados por 2 o más palabras.

Por otro lado, Stanford Core NLP es utilizado para las siguientes tareas: tokenización de palabras, segmentación de oraciones, etiquetado de parte de discurso (POS) y análisis estructural (análisis de árbol de circunscripción). Sin embargo Stanford Core NLP no trabaja con nombres conformados por 2 o más palabras, considerando estas palabras que conforman un nombre, como nombres de entidades distintas, lo cual generara tripletas innecesarias al momento de la transformación, siendo este problema resuelto por el uso de Senna.

La idea es usar Senna para el reconocimiento de todos los nombres que sean parte de un texto, sin distinguir si algunos nombres tienen 2 o más palabras. Senna es una muy buena herramienta para este proceso, y además nos retorna un archivo con cada palabra y su función gramatical en el texto, es por ello que es fácil reconocer los nombres por medio de un script ya que tiene su etiqueta particular. Entonces si encontramos 2 o más palabras que tienen la etiqueta de un nombre, lo unimos, de esta manera conseguimos aprovechar a Senna para esta parte del pre-procesamiento.

5.2.2. Extracción de conocimiento

Existen diferentes métodos para poder extraer conocimiento de un texto plano, uno de ellos es utilizando redes neuronales con modelos entrenados o pre-entrenados. En nuestra arquitectura utilizamos un enfoque de reconocimiento y extracción de entidades nombradas que usan redes neuronales.

Es por ello que en esta sección, que es la más importante de toda nuestra estructura propuesta, utilizamos el método propuesto en [Giorgi et al., 2019] por Giorgio et al., en donde se presenta un modelo de red neuronal de extremo a extremo para extraer entidades de forma conjunta y sus relaciones, que no se basa en herramientas de NLP externas y que integra un modelo de lenguaje extenso y previamente entrenado. Este enfoque permite ingresar oraciones de las que posteriormente extraerá y reconocerá las entidades, entonces con nuestra salida del pre-procesamiento, puede ya ser la entrada para esta fase y este algoritmo.

Este método es elegido debido a que es bastante actual en el estado del arte, además muestra mejores resultados en las comparaciones con respecto a sus similares y realiza ambas tareas, es decir reconocer y extraer entidades.

En nuestra implementación actual buscamos tener un flujo completo y por módulos de cada fase del enfoque y para ello se utiliza NLTK para el reconocimiento de entidades nombradas, sin embargo esta herramienta no es la más eficiente para esta tarea, por otra parte Spotlight de python es utilizado para la extracción de entidades y vinculación con la web, esta herramienta nos permite vincular las entidades con DBpedia, el cual está basado en los datos de wikipedia. Con esto logramos tener un flujo completo en la parte de extracción del conocimiento, aunque como ya mencionamos se podría mejorar en algunas tareas con enfoques más especializados.

5.2.3. Representación de datos

Una parte de esta fase final de nuestro enfoque es determinar los roles que cumplen las entidades en los distintos textos, es por ello que utilizamos la herramienta SRL MateTools, esta herramienta nos permite etiquetar roles semántica dentro de textos, también tiene otras funcionalidades sin embargo el etiquetado de roles nos interesa en nuestro enfoque. Finalmente en esta etapa, se realiza también la serialización de los datos en nuestro caso tripletas y para realizar esto utilizaremos la herramienta python rdflib que nos permite crear tripletas y serializarlo en diferentes formatos como XML/RDF, Json LD, Turtle, Ntriple.

5.3. Experimentos

5.3.1. Pruebas en la fase de preprocesamiento

Para realizar los experimentos, se ha seleccionado 13 fragmentos de distintas noticias en las que existen 30 casos donde hay nombres de entidades conformados por 2 o más palabras. A continuación se muestra los resultados obtenidos.

Como ejemplo mostraremos 3 oraciones en donde se puede mostrar la diferencia entre las entradas y salidas.

Oración 1: *Andrew Morris, managing consultant at Turnkey Consulting, says: AI and machine learning are of strategic importance.*

Salida: *AndrewMorris, managing consultant at TurnkeyConsulting, says : AI and machine learning are of strategic importance.*

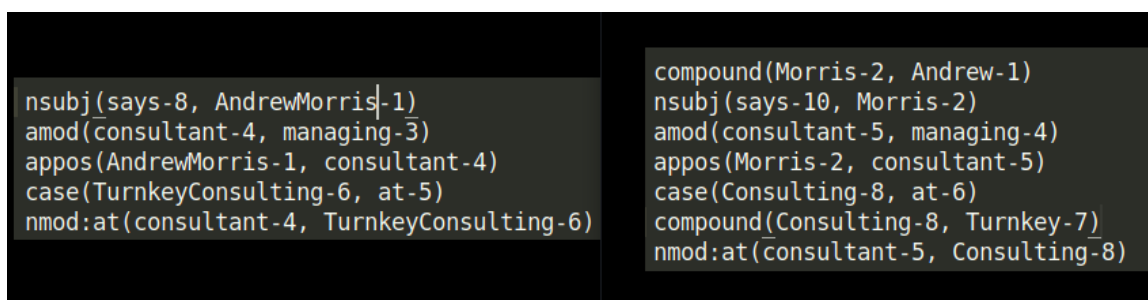


Figura 5.1: Resultados de StanfordCoreNLP para la oración 1, utilizando Senna en la imagen de la izquierda

Oración 2: *Forrester Research predicts that the UK is likely to come out of the crisis in a worse state than European competitors.*

Salida: *ForresterResearch predicts that the UK is likely to come out of the crisis in a worse state than European competitors.*

Oración 3: *One recent sign of this is a new job role, as reported in The Guardian*

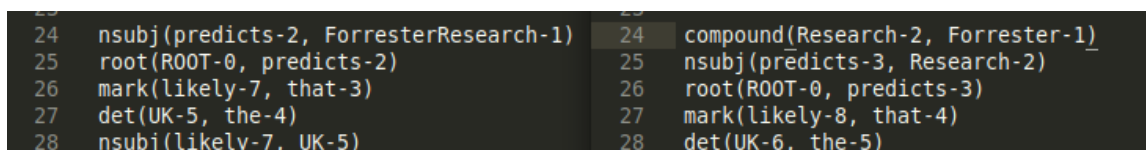


Figura 5.2: Resultados de StanfordCoreNLP para la oración 2, utilizando Senna en la imagen de la izquierda

Tabla 5.1: Cantidad de nombres de entidades unidos correcta e incorrectamente.

N° de Oración	1	2	3	4	5	6	7	8	9	10	11	12	13	Total
Nombres Unidos Correctos	2	1	2	1	1	2	2	4	2	3	2	3	3	28
Nombres Unidos Incorrectos	0	0	0	0	0	1	0	0	0	0	0	1	0	2
Total	2	1	2	1	1	3	2	4	2	3	2	4	3	30

on 11 July, for the head of a Downing Street data analytics unit, to be known as 10 data science or 10ds.

Salida: One recent sign of this is a new job role, as reported in TheGuardian on 11 July, for the head of a DowningStreet data analytics unit, to be known as 10 data science or 10ds.

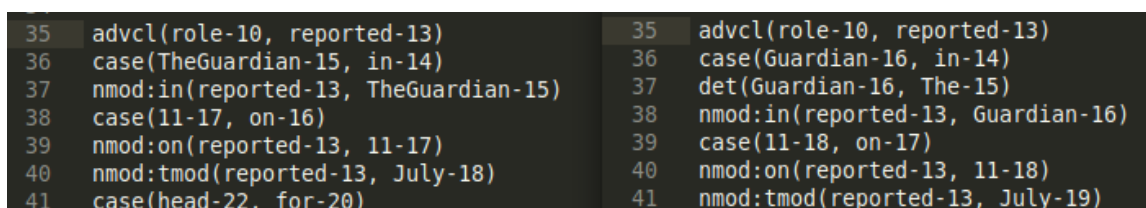


Figura 5.3: Resultados de StanfordCoreNLP para la oración 3, utilizando Senna en la imagen de la izquierda

Si observamos los resultados, en las Figuras 5.1, 5.2, y 5.3, se muestran resultados obtenidos de StanfordCoreNLP utilizando Senna (imagen de la izquierda) y sin usarlo (imagen de la derecha). Claramente se puede observar como StanfordCoreNLP considera a los nombre con mas de 2 palabra como si fueran distintas y atribuye sus acciones solo a uno de ellos. Por otro lado, nuestra propuesta que utiliza el análisis de Senna, une los nombres con 2 o mas palabras en una sola y esto ocurre de forma correcta en los 3 ejemplos utilizados. Sin embargo, existen casos en los que no existe una correcta unión.

En La Tabla 5.1 se presenta un resumen de las 13 oraciones seleccionadas y la cantidad de nombres unidos correctamente e incorrectamente. Como se observa, el número de aciertos es bastante alto en comparación al número de nombres que no han sido unidos, lo que significa que es un buen punto de partida para poder realizar los siguientes pasos en nuestro enfoque.

5.3.2. Pruebas para la fase de extracción de conocimiento

Como habíamos mencionado anteriormente, este enfoque también utiliza otros modelos para la fase de extracción de conocimiento, en contraste con enfoques similares. Es por ello que es fundamental realizar pruebas en esta etapa.

Para ello utilizamos como conjunto de datos, los datos de BBC, que consta de 5000 documentos o noticias sobre distintos temas, mencionar además que este conjunto de datos es utilizado por diferentes enfoques que realizan tareas similares.

En nuestro enfoque, utilizamos Spacy para la tarea de reconocimiento de entidades nombradas, esto representa una parte del procedimiento de esta etapa, posteriormente utilizamos Spotligh para la vinculación de entidades con recursos de la web.

Algunos ejemplos específicos se muestran a continuación:

Oración 1: *It has already offered to pay \$300m to settle charges, in a deal that is under review by the SEC.*

Salida:

```
xml <?xmlversion = "1.0" encoding = "UTF - 8"? >< rdf : RDF
xmlns : foaf = "http : //xmlns.com/foaf/0.1/"
xmlns : rdf = "http : //www.w3.org/1999/02/22 - rdf - syntax - ns" >
< rdf : Descriptionrdf : about = "http : //dbpedia.org/resource/SoutheasternConference" ><
rdf : typerdf : resource = "http : //xmlns.com/foaf/0.1/Person" / >
< foaf : name > SEC</foaf : name >< /rdf : Description >< /rdf : RDF >
```

Oración 2: *"For 2005, TimeWarner is projecting operating earnings growth of around 5, and also expects higher revenue and wider profit margins."*

Salida:

```
xml <?xmlversion = "1.0" encoding = "UTF - 8"? >< rdf : RDFxmlns : foaf =
"http : //xmlns.com/foaf/0.1/"
xmlns : rdf = "http : //www.w3.org/1999/02/22 - rdf - syntax - ns" >
< rdf : Descriptionrdf : about = "http : //dbpedia.org/resource/WarnerMedia" ><
rdf : typerdf : resource = "http : //xmlns.com/foaf/0.1/Person" / >
< foaf : name > TimeWarner</foaf : name >< /rdf : Description >< /rdf :
RDF >
```

Oración 3: *"However, the company said AOL's underlying profit before exceptional items rose 8 on the back of stronger internet advertising revenues."*

Salida:

```
xml <?xmlversion = "1.0" encoding = "UTF - 8"? >< rdf : RDFxmlns : foaf =
"http : //xmlns.com/foaf/0.1/"xmlns : rdf = "http : //www.w3.org/1999/02/22 -
rdf - syntax - ns" >
< rdf : Descriptionrdf : about = "http : //dbpedia.org/resource/AOL" >
```

```
<rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person" />
<foaf:name>8</foaf:name></rdf:Description></rdf:RDF>
```

Como podemos observar en las 3 oraciones las entidades que se encontraron también fueron vinculadas a un recurso en la web, en el caso de FRED podemos comprobar que algunas de estas entidades no fueron vinculadas a recursos de la Web, lo que significa que en esos términos, nuestro enfoque es mejor.

También se pudo comprobar que FRED al momento de ingresar oraciones de tamaño regular, demora demasiado en devolver un resultado, en el caso de nuestro enfoque al momento del preprocesamiento, donde se divide en oraciones mejora este proceso, ya que esto flexibiliza el proceso y lo acelera.

Capítulo 6

Conclusiones

La conversión de texto a RDF es fundamentales en la Web Semántica y en el mundo actual, ya que tiene muchas aplicaciones como por ejemplo en la recuperación de información mucho más precisa. Desde hace muchos años toda esta información se ha ido almacenando en bases de datos relaciones o archivos XML, HTML y sobre todo en forma de texto desde los inicio de la Web. Es por ello la importancia y la necesidad de convertir estos datos a RDF y aprovechar de mejor manera toda esta información.

Este campo de investigación es relativamente nuevo, sin embargo existen diversos enfoque desarrollados hasta el momento con la peculiaridad de dividir el problema en 3 partes fundamentales, que por sus autores son definidos con diferentes nombres. Estas 3 partes son, el pre-procesamiento, la extracción de conocimiento y la representación de los datos.

En nuestro enfoque pretendemos mejorar tanto en la fase de pre-procesamiento y la fase de extracción de conocimiento, añadiendo nuevas herramientas para solucionar o mejorar la conversión. En el pre-procesamiento, utilizamos Senna para unir nombres que tengan 2 o más palabras, esto debido a que StanfordCoreNLP, utilizado también en nuestro enfoque, considera a los nombre con 2 o más palabras como si fueran distintos, lo que genera tripletas repetidas o innecesarias. En la parte de extracción de conocimiento utilizamos Spacy de python para el reconocimiento de entidades, por otro lado utilizamos spotlight para vincular la entidades con recursos de la web.

Las pruebas realizadas para la etapa de pre-procesamiento, demuestran que el uso de Senna para la unión de nombre da resultados mejores en comparación a la propuesta presentado por Martinez en [Martinez-Rodriguez et al., 2019a], ya que de 30 nombre seleccionados de un grupo de noticias de TI, 28 fueron unidas correctamente, mientras que solo 2 no se unieron, lo que significa que los resultados son alentadores y se puede continuar con la siguiente fase. Además, presentamos 3 oraciones en las que se puede ver cual es la salida de StanfordCoreNLP utilizando y sin utilizar Senna, donde se puede ver claramente que si no usamos Senna solo una palabra de los nombre con varias palabras es considerado.

Las pruebas realizadas en la etapa de extracción de conocimiento, demuestran que nuestro enfoque es mucho mas preciso al momento de reconocer y vincular entidades con

la web, en contraste con FRED. Además de que el tiempo de ejecución es menor al dividir en oraciones la entrada de texto.

En la fase de extracción del conocimiento se ha logrado el reconocimiento de entidades nombradas con Spacy y la extracción y vinculación con Spotlight, ambas herramientas de python, de esta manera logramos encontrar entidades y lo vinculamos con la web, actualmente buscamos recursos en DBpedia que está basado en wikipedia. Contamos con una interfaz web capaz de recibir como entrada un texto plano y transformarlo a RDF, además para observar los resultados podemos seleccionar el formato de serialización que nos guste.

6.1. Trabajos futuros

Nuestro principal objetivo en adelante es mejorar la fase de extracción de conocimiento con enfoques que sean más especializados y además que obtengan mejores resultados en el reconocimiento de entidades nombradas. Por otra parte en la fase de extracción y vinculación de entidades también es fundamental utilizar un enfoque que busque otras fuentes de la web para obtener ubicación de recursos y vincularlos a nuestras entidades. De esta forma nuestro enfoque tendría mejores resultados con mejor consistencia y menor redundancia que es lo que buscamos.

Bibliografía

- [Thu,] Dtd2owl: Transformación automática de documentos xml en owl ontology. In ACM, editor, *Actas de la 2ª Conferencia Internacional sobre Ciencias de la Interacción: Tecnología de la Información, Cultura y Humanos*, ICIS '09.
- [Zak, 2019] (2019). Extracción de representación semántica de texto árabe no estructurado.
- [Abdelaal, 2019] Abdelaal, H. (2019). Knowledge extraction from simplified natural language text.
- [Augenstein et al., 2012] Augenstein, I., Padó, S., and Rudolph, S. (2012). Lodifier: Generating linked data from unstructured text. *Semantic Web: Res Appl*, 7295:210–224.
- [Berners-Lee, 2009] Berners-Lee, T. (2009). Relational databases on the semantic web. Online; accessed 2019-07-01.
- [Berners-Lee and Hendler, 2001] Berners-Lee, T. and Hendler, J. (2001). Publishing on the semantic web. *Nature*, 410:1023–4.
- [Bizer et al., 2018] Bizer, C., Vidal, M.-E., and Weiss, M. (2018). *RDF Technology*, pages 3106–3109. Springer New York, New York, NY.
- [Blog, 2018] Blog, B. (2018). The bulk of data wandering on the net is unstructured data. <https://blog.bitext.com/the-bulk-of-data-wandering-on-the-net-is-unstructured-data>. Online; accessed 2020-05-25.
- [Breitling, 2009a] Breitling, F. (2009a). A standard transformation from xml to rdf via xslt. *Astronomische Nachrichten*, 330.
- [Breitling, 2009b] Breitling, F. (2009b). A standard transformation from xml to rdf via xslt. *Astronomische Nachrichten*, 330.
- [Carothers and Seaborne, 2014] Carothers, G. and Seaborne, A. (2014). Rdf 1.1 n-triples. Online; accessed 2020-05-14.
- [David Beckett and Machina, 2014] David Beckett, Tim Berners-Lee, E. P. G. C. and Machina, L. (2014). Rdf 1.1 turtle. Online; accessed 2020-05-14.
- [Deursen et al., 2008a] Deursen, D. V., Poppe, C., Martens, G., Mannens, E., and d. Walle, R. V. (2008a). Xml to rdf conversion: A generic approach. In *2008 International*

Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution, pages 138–144.

- [Deursen et al., 2008b] Deursen, D. V., Poppe, C., Martens, G., Mannens, E., and d. Walle, R. V. (2008b). Xml to rdf conversion: A generic approach. In *2008 International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution*, pages 138–144.
- [Draicchio et al., 2013] Draicchio, F., Gangemi, A., Presutti, V., and Nuzzolese, A. (2013). Fred: From natural language text to rdf and owl in one click. 7955:263–267.
- [Duerst, 2004] Duerst, M. (2004). Internationalized resource identifiers (iri). <https://www.w3.org/International/iri-edit/draft-duerst-iri.html>. Online; accessed 2020-05-25.
- [Ermilov et al., 2013] Ermilov, I., Auer, S., and Stadler, C. (2013). Csv2rdf: User-driven csv to rdf mass conversion framework.
- [Exner and Nugues, 2012] Exner, P. and Nugues, P. (2012). Entity extraction: From unstructured text to dbpedia rdf triples. *CEUR Workshop Proceedings*, 906.
- [Gandon and Schreiber, 2014] Gandon, F. and Schreiber, G. (2014). Rdf 1.1 sintaxis xml. Online; accessed 2020-05-14.
- [Gangemi et al., 2013] Gangemi, A., Draicchio, F., Presutti, V., Nuzzolese, A. G., and Reforgiato, D. (2013). A machine reader for the semantic web. page 149–152.
- [Giorgi et al., 2019] Giorgi, J., Wang, X., Sahar, N., Shin, W., Bader, G., and Wang, B. (2019). End-to-end named entity recognition and relation extraction using pre-trained language models.
- [Graham Klyne, 2014] Graham Klyne, Jeremy J. Carroll, B. M. (2014). Rdf 1.1 concepts and abstract syntax. <https://www.w3.org/TR/rdf11-concepts/#section-Graph-Literal>. Online; accessed 2020-05-26.
- [Graham Klyne and McBride, 2014] Graham Klyne, J. J. C. and McBride, B. (2014). Rdf 1.1 concepts and abstract syntax. Online; accessed 2020-05-14.
- [Group, 2014] Group, R. W. (2014). Resource description framework (rdf). Online; accessed 2020-05-14.
- [Group, 2013] Group, W. S. W. (2013). Sparql 1.1 overview. <https://www.w3.org/TR/sparql11-overview/>. Online; accessed 2020-05-26.
- [GRUBER., 1993] GRUBER., T. R. (1993). *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. PhD thesis, Knowledge Systems Laboratory, Stanford University, CA,.
- [Hassanzadeh et al., 2013] Hassanzadeh, K., Reformat, M., Pedrycz, W., Jamal, I., and Berezowski, J. (2013). T2r: System for converting textual documents into rdf triples. pages 221–228.

-
- [Hazber et al., 2016] Hazber, M. A. G., Li, R., Xu, G., and Alalayah, K. M. (2016). An approach for automatically generating r2rml-based direct mapping from relational databases. In Che, W., Han, Q., Wang, H., Jing, W., Peng, S., Lin, J., Sun, G., Song, X., Song, H., and Lu, Z., editors, *Social Computing*, pages 151–169, Singapore.
- [Hert et al., 2011] Hert, M., Reif, G., and Gall, H. C. (2011). A comparison of rdb-to-rdf mapping languages. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, pages 25–32, New York, NY, USA. ACM.
- [Holzinger et al., 2013] Holzinger, A., Stocker, C., Ofner, B., Prohaska, G., Brabenetz, A., and Hofmann-Wellenhof, R. (2013). *Combining HCI, Natural Language Processing, and Knowledge Discovery - Potential of IBM Content Analytics as an Assistive Technology in the Biomedical Field*, pages 13–24.
- [Huang et al., 2015] Huang, J.-Y., Lange, C., and Auer, S. (2015). Streaming transformation of xml to rdf using xpath-based mappings. In *Proceedings of the 11th International Conference on Semantic Systems*, pages 129–136, New York, NY, USA. ACM.
- [Lapiente, 2013] Lapiente, M. J. L. (2013). Hipertexto, el nuevo concepto de documento en la cultura de la imagen.
- [Laufer, 2015] Laufer, C. (2015). Web semántica. Online; accessed 2020-05-14.
- [Mahmud et al., 2018] Mahmud, S. M. H., Hossin, M., Jahan, H., Noori, S., and Hossain, M. (2018). Csv2rdf: Generating rdf data from csv file using semantic web technologies. *Journal of Theoretical and Applied Information Technology*, 96:6889–6902.
- [Manu Sporny and Lindström, 2014] Manu Sporny, Dave Longley, G. K. M. L. and Lindström, N. (2014). A json-based serialization for linked data. Online; accessed 2020-05-14.
- [Martinez-Rodriguez et al., 2019a] Martinez-Rodriguez, J., Lopez-Arevalo, I., Rios-Alvarado, A., Hernandez, J., and Aldana-Bobadilla, E. (2019a). Extraction of rdf statements from text. pages 87–101.
- [Martinez-Rodriguez et al., 2019b] Martinez-Rodriguez, J., Lopez-Arevalo, I., Rios-Alvarado, A., Hernandez, J., and Aldana-Bobadilla, E. (2019b). Extraction of rdf statements from text. pages 87–101.
- [Michel et al., 2014] Michel, F., Montagnat, J., and Faron-Zucker, C. (2014). A survey of rdb to rdf translation approaches and tools.
- [Patrick J. Hayes, 2014] Patrick J. Hayes, Florida IHMC, P. F. P.-S. (2014). Rdf 1.1 semantics. <https://www.w3.org/TR/2014/REC-rdf11-mt-20140225/#blank-nodes>. Online; accessed 2020-05-26.
- [Salas et al., 2011] Salas, P. E., Marx, E., Mera, A., and Viterbo, J. (2011). Rdb2rdf plugin: Relational databases to rdf plugin for eclipse. In *Proceedings of the 1st Workshop on Developing Tools As Plug-ins*, TOPI '11, pages 28–31, New York, NY, USA. ACM.
- [Schultz, 2019] Schultz, J. (2019). How much data is created on the internet each day? <https://blog.microfocus.com/how-much-data-is-created-on-the-internet-each-day/>. Online; accessed 2020-05-25.

-
- [Stadler et al., 2015] Stadler, C., Unbehauen, J., Westphal, P., Sherif, M., and Lehmann, J. (2015). Simplified rdb2rdf mapping. volume 1409.
- [Verma et al., 2020] Verma, S., Jain, K., and Prakash, C. (2020). An unstructured to structured data conversion using machine learning algorithm in internet of things (iot).
- [Vidal et al., 2014] Vidal, V. M. P., Casanova, M. A., Neto, L. E. T., and Monteiro, J. M. (2014). A semi-automatic approach for generating customized r2rml mappings. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing, SAC '14*, pages 316–322, New York, NY, USA. ACM.
- [W3, 2016] W3 (2016). Rdf schema (rdfs). <https://www.w3.org/wiki/RDFS>. Online; accessed 2019-09-27.