

RHEINISCHE FRIEDRICH-  
WILHELMS-UNIVERSITÄT BONN  
INSTITUT FÜR INFORMATIK III

MASTER THESIS

---

# Automated Link Discovery for Data Harmonization in the Maritime Domain

---

Jaime Manuel Trillos Ujueta  
E-mail: trillosj@informatik.uni-bonn.de

1. *Examiner:* Prof. Dr. Jens Lehmann

2. *Examiner:* Dr. Damien Graux

*Supervisors:* Dr. Hajira Jabeen - Dr. Ioanna Lytra

*A thesis submitted in fulfillment of the requirements for the  
degree of Master of Science*

*in the*

Smart Data Analytics  
Institute of Computer Science

April 10, 2019





# Declaration of Authorship

I, JAIME MANUEL TRILLOS UJUETA, declare that this thesis titled, "Automated Link Discovery for Data Harmonization in the Maritime Domain" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

---

Date:

---



## *Abstract*

Faculty of Mathematics and Natural Science  
Institute of Computer Science

Master of Science

### **Automated Link Discovery for Data Harmonization in the Maritime Domain**

by Jaime Manuel Trillos Ujueta

There is a large amount of heterogeneous data sources available for the maritime domain; those data sources are very specialized that only experts can understand them. Additionally, there are numerous methods for extracting, processing and interlinking data with linked open data sources (LOD). Consequently, those methods transform the data sources in linked open vocabularies and publish them in online repositories to later connect with other vocabularies and ontologies. In this thesis, we present an alternative approach that allows extracting the metadata from maritime data sources for linking with linked open vocabularies using Natural Language Processing (NLP) for Named Entity Recognition (NER) and interlinking tools called Automated Link Discovery process. We implement the Automated Link Discovery process using Stanford Named Entity Recognizer and LIMES tools in the BigDataOcean Harmonization Tool for the automated extraction of entities of heterogeneous data sources metadata, such as Copernicus, NetCDF, CSV, and Excel, and linking them with the vocabularies from the BigDataOcean Vocabulary Repository. Thus, we create a gold standard to evaluate the accuracy of using the two tools using precision, recall, and F-measure. Furthermore, using the gold standard, we assessed the performance of the Automated Link Discovery. Consequently, it is possible to conclude that the Automated Link Discovery is the correct solution for filling missing metadata information given by the dataset, and it requires less expert support to insert new dataset metadata into the BigDataOcean Harmonization Tool.



# *Acknowledgements*

I would like to thank Dr. Hajira Jabeen and Dr. Ioanna Lytra for their support and guidance through the development of the BigDataOcean Harmonization tool and this master thesis.





# Contents

<b>Declaration of Authorship</b>	<b>ii</b>
<b>Abstract</b>	<b>iv</b>
<b>Acknowledgements</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background</b>	<b>4</b>
2.1 Metadata . . . . .	4
2.2 Semantic Web . . . . .	4
2.2.1 RDF . . . . .	5
2.2.2 SPARQL . . . . .	5
2.2.3 Vocabularies . . . . .	6
2.2.4 Linked Open Data . . . . .	7
2.3 Interlinking . . . . .	8
2.3.1 Entity Linking . . . . .	8
2.3.2 Evaluation Measures for Interlinking . . . . .	8
2.3.2.1 Precision . . . . .	9
2.3.2.2 Recall . . . . .	9
2.3.2.3 F1 Score . . . . .	9
<b>3 State of the Art</b>	<b>10</b>
3.1 Related Literature . . . . .	10
3.2 Related Tools . . . . .	12
3.2.1 Natural Language Processing for Named Entity Recognition . . . . .	12
3.2.1.1 Stanford Named Entity Recognizer . . . . .	13
3.2.1.2 Apache OpenNLP . . . . .	16
3.2.1.3 Comparison between Stanford NER and Apache OpenNLP . . . . .	17
3.2.2 Interlinking Tools . . . . .	19
3.2.2.1 LIMES . . . . .	20
3.2.2.1.1 Metadata . . . . .	21
3.2.2.1.2 Prefixes . . . . .	21
3.2.2.1.3 Data Sources . . . . .	21
3.2.2.1.4 Metrics . . . . .	22
3.2.2.1.5 Conditions . . . . .	25
3.2.2.1.6 Output . . . . .	25
3.2.2.2 Silk . . . . .	26

3.2.2.3	Comparison between LIMES and Silk . . . . .	29
<b>4</b>	<b>BigDataOcean Project</b>	<b>31</b>
4.1	BigDataOcean Vocabulary Repository . . . . .	32
4.2	BigDataOcean Harmonization Tool . . . . .	34
<b>5</b>	<b>Methodology</b>	<b>39</b>
5.1	Architecture Implementation . . . . .	40
5.2	Automated Link Discovery Implementation . . . . .	43
5.2.1	Stanford Named Entity Recognizer . . . . .	44
5.2.1.1	Post-processing of Stanford NER . . . . .	44
5.2.1.2	Training Data . . . . .	45
5.2.2	LIMES . . . . .	47
5.2.2.1	Data Sources . . . . .	47
5.2.2.2	Selection of Metrics . . . . .	48
5.2.2.3	Acceptance and Review Conditions . . . . .	49
5.3	Front-end Implementation . . . . .	50
<b>6</b>	<b>Evaluation</b>	<b>53</b>
6.1	Datasets . . . . .	53
6.2	Evaluation of String Similarity Measures . . . . .	54
6.3	Accuracy Evaluation . . . . .	58
6.3.1	Accuracy Evaluation with or without Post-processing Method before Stanford NER . . . . .	58
6.3.2	Accuracy Evaluation with Post-processing Method using Stanford NER and LIMES . . . . .	60
6.4	Performance Evaluation . . . . .	64
<b>7</b>	<b>Conclusions</b>	<b>67</b>
	<b>Bibliography</b>	<b>69</b>

# List of Figures

2.1	Metadata example . . . . .	4
2.2	RDF Triple structure . . . . .	5
2.3	Part of the Linking Open (LOD) Data Project Cloud Diagram [1] . . . .	7
2.4	Essential four terms for evaluation measures [2] . . . . .	9
3.1	Stanford NER GUI tool . . . . .	14
3.2	LIMES GUI . . . . .	20
3.3	Example of Silk-LSL . . . . .	26
3.4	Workspace browser component of Silk Workbench . . . . .	27
3.5	Linkage rule component of Silk Workbench . . . . .	28
3.6	Evaluation component of Silk Workbench . . . . .	28
4.1	BigDataOcean Vocabulary Repository. . . . .	32
4.2	List of APIs which BigDataOcean Vocabulary Repository offers. . . . .	33
4.3	JSON file of the result from executing the Info Vocab API of Canonical Model Vocabulary. . . . .	34
4.4	Flowchart Diagram BigDataOcean Harmonization Tool. . . . .	36
4.5	BigDataOcean Harmonization Dataset Metadata Structure Model. . . . .	36
5.1	Methodology workflow diagram . . . . .	39
5.2	Pipeline Diagram for the creation of the Automated Link Discovery process. . . . .	40
5.3	Example of Pipeline Diagram for the creation of the Automated Link Discovery process. . . . .	41
5.4	BigDataOcean Harmonization Tool class diagram. . . . .	42
5.5	Architecture Diagram. . . . .	43
5.6	Training Data of Named Entity Recognition. . . . .	45
5.7	Training Data of Named Entity Recognition Flow Chart. . . . .	46
5.8	Flowchart diagram of BigDataOcean Harmonization Tool using Vocabu- lary Repository, Stanford NER and LIMES . . . . .	51
5.9	Pending dataset files to be added into Harmonization Tool. . . . .	51
5.10	Snippet of the frontend where metadata suggested using Stanford NER and LIMES . . . . .	52
5.11	Snippet of dataset metadata saved in BigDataOcean Harmonization Tool . . . . .	52
6.1	Representation of the gold standard . . . . .	54
6.2	Stanford NER post-processing behavior . . . . .	58
6.3	Total number of entities, number of subjects, number of keywords and number of geographic locations using Stanford NER and LIMES only for Copernicus datasets . . . . .	61

6.4	Total number of entities, number of subjects, number of keywords and number of geographic locations using Stanford NER and LIMES . . . . .	62
6.5	Number of raw variables and Number of linked variables using LIMES . .	63
6.6	Performance evaluation for subject, keywords, and geographic location using Stanford NER and LIMES only for Copernicus datasets . . . . .	64
6.7	Performance evaluation for subject, keywords, and geographic location using Stanford NER and LIMES . . . . .	65
6.8	Performance evaluation for variables using LIMES . . . . .	65

# List of Tables

3.1	Stanford Named Entity Recognizer classifiers . . . . .	13
3.2	Comparison between Stanford NER and Apache OpenNLP . . . . .	18
3.3	LIMES Measure packages. . . . .	23
3.4	Silk measure packages . . . . .	28
3.5	Comparison between Limes and Silk . . . . .	29
4.1	Harmonization dataset metadata linked with vocabularies. . . . .	38
5.1	BigDataOcean Harmonization Tool models classifiers . . . . .	44
5.2	BigDataOcean Harmonization Tool models classifiers . . . . .	44
6.1	Gold standard dataset . . . . .	55
6.2	Subjects string similarity measure evaluation . . . . .	56
6.3	Keywords string similarity measure evaluation . . . . .	56
6.4	Geographic location string similarity measure evaluation . . . . .	57
6.5	Variables string similarity measure evaluation . . . . .	57
6.6	Linked results given by LIMES using entities with and without post- processing (S = Subject, K = Keywords, G = Geographic Location) . . .	59
6.7	Accuracy evaluation for the dataset “Copernicus 1” . . . . .	60
6.8	Accuracy evaluation for the dataset “NetCDF 3” . . . . .	60
6.9	Accuracy evaluation for the dataset “Excel” . . . . .	60
6.10	Canonical named variables for the dataset for the dataset “NetCDF 3” linked with raw variables using LIMES . . . . .	63
6.11	Accuracy evaluation for the dataset “Copernicus 1” with Cosine similarity measure . . . . .	63
6.12	Accuracy evaluation for the dataset “Copernicus 1” with Trigrams measure	64
6.13	Accuracy evaluation for the dataset “Copernicus 1” with Qgrams measure	64

# List of Listings

2.1	RDF triple example . . . . .	5
2.2	SPARQ query example . . . . .	6
3.1	Command line for printing named entities . . . . .	14
3.2	Command line for printing named entities in a TSV file . . . . .	14
3.3	Command line for printing named entities using the three classifier models . . . . .	15
3.4	A snippet code from the API code presented by Stanford NER . . . . .	15
3.5	Command line for printing named entities using Apache OpenNLP . . . . .	16
3.6	Example of Command line for printing named entities using Apache OpenNLP . . . . .	17
3.7	Output of the listing 3.6 . . . . .	17
3.8	A snippet of the API code . . . . .	17
3.9	Metadata of LIMES configuration file . . . . .	21
3.10	Prefix of LIMES configuration file . . . . .	21
3.11	Data Sources of LIMES configuration file . . . . .	22
3.12	Metrics of LIMES configuration file . . . . .	25
3.13	Conditions of LIMES configuration file . . . . .	25
3.14	Output tag of LIMES configuration file . . . . .	25
4.1	A snippet of the Med Sea - NRT in situ Observations dataset metadata in Turtle format . . . . .	37
5.1	Snippet of training configuration . . . . .	46
5.2	LIMES target data configuration example . . . . .	48
5.3	LIMES source data configuration for the pipeline variables . . . . .	48
5.4	LIMES source data configuration for the pipeline subjects . . . . .	49
5.5	Metric configuration definition in LIMES for variables . . . . .	49
5.6	Acceptance condition in LIMES for variables pipeline . . . . .	49
5.7	Review condition in LIMES for variables pipeline . . . . .	50



*Dedicated to my parents and sister. . .*





# Chapter 1

## Introduction

The semantic web is data web [3] which aims to integrate data in RDF (Resource Description Framework) format which allows being shared and reused across different data content, information applications, and systems. However, a vital problem in Semantic Web is the capability to match, compare, and resolve data referring to the same real-world objects in form of data links. Data linking or Linked Data is the mechanism for publishing semantic data on the web allowing to interlink two object descriptions that refer to the same real-world object for a given domain and become more useful through semantic queries. Linked Open Data<sup>1</sup> (LOD) promotes the idea of interlinking heterogeneous data sources, under the assumption of the degree of similarity between two data sources descriptions. The higher the probability of similarity degree is, it can refer to the same object.

There is a large amount of heterogeneous data sources available for the maritime domain; those data sources are very specialized that only experts can understand them. Additionally, there are numerous methods for extracting, processing and interlinking data with LOD sources. Consequently, those methods transform and publish the data sources in linked open vocabularies to connect with other vocabularies/ontologies. In this thesis, we present a new approach that allows extracting the metadata from maritime data sources for interlinking with linked open vocabularies using Natural Language Processing (NLP) for Named Entity Recognition (NER) and interlinking tools.

BigDataOcean project aims to enable maritime big data scenarios for the European Union through a multi-segment platform that will combine data from different data sources, and volume under an interlinked trusted multilingual engine [4]. The BigDataOcean project offers tools as BigDataOcean Vocabulary Repository for accessing

---

<sup>1</sup><https://lod-cloud.net/>

and sharing Linked Big Data vocabularies and ontologies related to the maritime domain, and BigDataOcean Harmonization Tool for extraction of metadata from datasets written in different formats, such as NetCDF, CSV, and Excel; as well as datasets available by external services, like the Copernicus service<sup>2</sup>.

Concerning Natural Language Processing for Named Entity Recognition, it is possible to find tools like Stanford Named Entity Recognizer tool and Apache OpenNLP. These tools were compared in the following aspects, the benefits, the number of available models, CLI applications, and RESTful API. In the case of interlinking data, there exist different tools; among them, we can find, Silk and LIMES. Same as the case before, we analyzed them by comparing the type of data sources, GUI applications, and RESTful API they offer.

The use of NLP improves semantic search results. Given that combining NLP, NER applications and Semantic Web support the automated extraction of relevant information from text, helping to link data to entities. Simultaneously, the use of LOD vocabularies enhances the semantic entities match. As a result, using interlinking tools provide matching semantic web vocabularies terms/instances with the metadata entities provided by the dataset.

This thesis is divided into two main parts:

- The comparison and selection of named entity recognition and interlinking tool that best suits the BigDataOcean Harmonization Tool.
- The development of the Automated Link Discovery process using vocabularies taken from the BigDataOcean Vocabulary Repository and the metadata extracted from the dataset given by the user.

In the second part, we present an automated process for interlinking metadata which is not explicitly given by the dataset metadata with the help of Automated Link Discovery. Automated Link Discovery will identify and match entities using the Stanford Named Entity Recognizer and LIMES tools. At the same time, the Automated Link Discovery will automatically recognize the canonical variables<sup>3</sup> for each raw variable<sup>4</sup> which is difficult to understand for a non-expert person, e.g., a raw variables is *dryt* and its canonical variable is *air\_temperature*.

This thesis aims:

---

<sup>2</sup><http://marine.copernicus.eu/services-portfolio/access-to-products/>

<sup>3</sup>Canonical variables means standard name given by BDO project or Climate and Forecast Standard Vocabulary

<sup>4</sup>The raw variables mean the variables taken from the dataset file

- To apply existing open source techniques in interlinking and named entity recognition to a new domain, marine data for the BigDataOcean project,
- To exploit the full potential of the BigDataOcean Harmonization Tool without using the expert input.

This thesis is structured as follows: Chapter 1 offers an introduction. Chapter 2 describes the technical terms associated with the thesis such as metadata, Semantic Web, and interlinking. In chapter 3, we overview related existing works. Additionally, we present the different tools for the Automated Link Discovery creation and the comparison of Natural Language Processing for Named Entity Recognition tools and also the comparison of interlinking tools. Chapter 4 presents the BigDataOcean project. Chapter 5 describes the integration process of BigDataOcean project with the automated linked discovery using Stanford Named Entity Recognizer and LIMES tools. Chapter 6 is divided into four sections; first, we present the gold standard dataset used for the accuracy and performance evaluation, second, we present the string similarity evaluation for LIMES, third, we present the accuracy evaluation when using the automated linked discovery, and last, we present the performance evaluation. Finally, Chapter 7 offers conclusions and future work.

## Chapter 2

# Background

### 2.1 Metadata

Metadata<sup>1</sup> means "data about data" which includes information about the attributes of other data but not the data itself. Figure 2.1 shows a metadata example of a publication. As seen in the figure, the publication metadata attributes are the title, abstract, authors, language, year and ISBN.

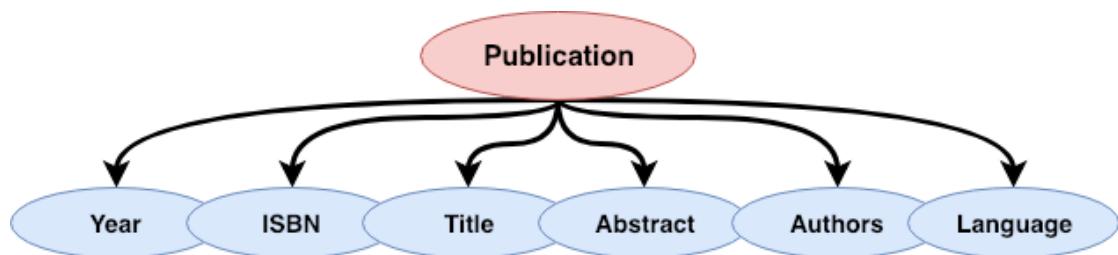


FIGURE 2.1: Metadata example

### 2.2 Semantic Web

The semantic web is about common formats for integration, and it is about language for recording how the data relates to real-world objects [5]. The semantic web allows a person or a machine to understand the meaning of the information distributed on the World Wide Web.

There are several standard notations or models to achieve the semantic web, such as RDF, RDFS, and OWL, which the main idea is to present a precise description of the concepts, terms, and relations within a given knowledge domain.

---

<sup>1</sup><https://en.wikipedia.org/wiki/Metadata>

### 2.2.1 RDF

Resource Description Framework<sup>2</sup> (RDF) is a standard model for data interchange on the Web. RDF originally used for metadata for web resources, which can be read by persons and machines, RDF is described in triples.

An RDF triple is a set of three entities which codifies a statement. Figure 2.2 presents the structure of a triple. The triple is divided into three parts:

**Subject:** describes the resource represented as a Uniform Resource Identifier (URI) or a blank node.

**Predicate:** refers to the property of a resource represented as URI.

**Object:** describes the value of the property, it can be represented as a URI, blank node or literal.

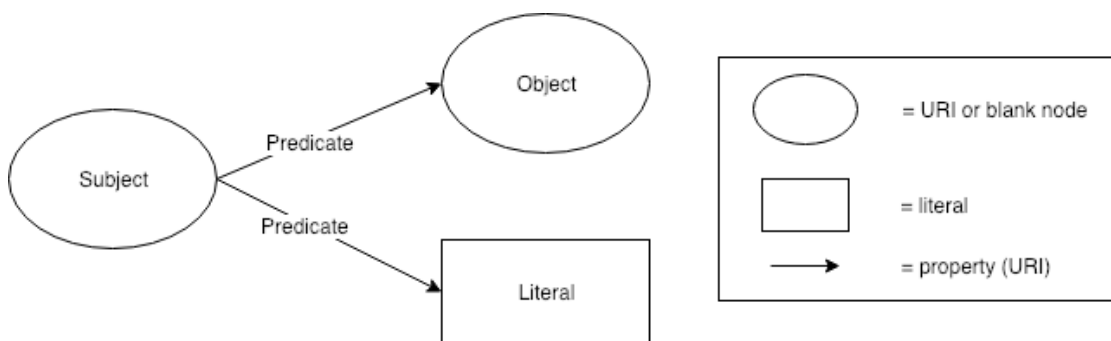


FIGURE 2.2: RDF Triple structure

An example of RDF triple can be “Spiderman is enemy of Green Goblin.” The following is the RDF representation:

```
<http://example.org/#spiderman>
<http://www.perceive.net/schemas/relationship/enemyOf> <http://
  example.org/#green-goblin> ;
<http://xmlns.com/foaf/0.1/name> "Spiderman" .
```

LISTING 2.1: RDF triple example

### 2.2.2 SPARQL

SPARQL<sup>3</sup> stands for “SPARQL Protocol and RDF Query Language”. SPARQL is a query language designed by the W3C RDF Data Access Working Group and protocol to query and manipulate RDF graph on the web or in RDF store.

<sup>2</sup><https://www.w3.org/2001/sw/wiki/RDF>

<sup>3</sup><https://www.w3.org/TR/sparql11-overview/>

SPARQL queries consist of triple patterns, conjunctions, disjunctions, and optional patterns. Listing 2.2 shows a simple example of a SPARQL query. This query retrieves the name of John's friend.

```
# prefix declarations
PREFIX ex: <http://example.com/resources/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

# query
SELECT ?friendname WHERE {
    ex:John foaf:knows ?friend .
    ?friend foaf:firstname ?friendname
}
```

LISTING 2.2: SPARQ query example

### 2.2.3 Vocabularies

The W3C defines the vocabularies as the concepts and terms (relationships) used to describe and represent an area of interest [6]. The difference between ontology and vocabulary is the level of abstraction and relationships among concept [7]. In other words, ontology is a vocabulary expressed in an ontology representation language (OWL<sup>4</sup>) which uses vocabulary terms to describe something in a specified area of interest, while vocabulary uses to establish a mechanism for allowing differing vocabularies to be mapped onto the ontology. The W3C defines the vocabularies as the concepts and terms (relationships) used to describe and represent an area of interest [6]. The difference between ontology and vocabulary is the level of abstraction and relationships among concept [7]. In other words, ontology is a vocabulary expressed in an ontology representation language (OWL<sup>5</sup>) which uses vocabulary terms to describe something in a specified area of interest, while vocabulary uses to establish a mechanism for allowing differing vocabularies to be mapped onto the ontology.

A vocabulary or ontology is represented as RDF model; as explained in section 2.2.1, RDF described as triples in which has a set of three entities: subject, predicate, and object. Vocabularies used the Resource Description Framework Schema<sup>6</sup> (RDFS) which allows defining classes, properties and restrict their use.

An example of a vocabulary is Friend of a Friend vocabulary<sup>7</sup> (foaf) that is devoted to linking people and information using the web. Foaf collects a variety of terms for

<sup>4</sup>OWL means Web Ontology Language <https://www.w3.org/OWL/>

<sup>5</sup>OWL means Web Ontology Language <https://www.w3.org/OWL/>

<sup>6</sup><https://www.w3.org/TR/rdf-schema/>

<sup>7</sup><http://xmlns.com/foaf/0.1/>

describing people (*foaf:person*), groups (*foaf:group*), or even documents (*foaf:document*). In the case of a foaf class is *Agent*, and for a foaf property is *currentProject*.

## 2.2.4 Linked Open Data

Linked Data is an approach method for publishing data to be interlinked and be useful through semantic queries [8]. Essentially, Linked Open Data (LOD) is on the web freely available with an open license written in an open standard format (RDF) which allows metadata to be interlinked to different data sources. Figure 2.3 shows a part of Linked Open Data. The goal of the semantic web is to interlink a large amount of data on the web in a way that can be read by a person and a machine. Tim Berners-Lee created four principles for helping users to upload a 5 five star Linked Data <sup>8</sup>:

- Use URIs as names for things
- Use HTTP URIs so that people can look up those names
- When someone looks up a URI, provide useful information, using the RDF standards
- Include links to other URIs so that they can discover more things.

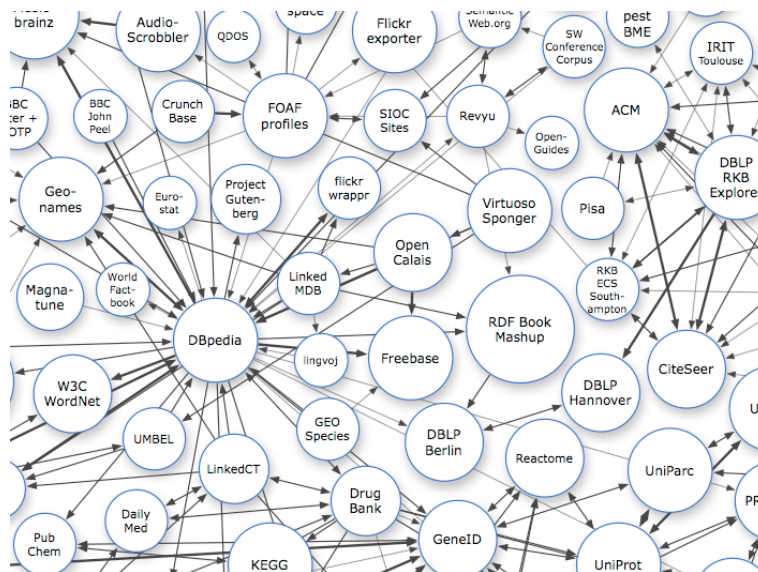


FIGURE 2.3: Part of the Linking Open (LOD) Data Project Cloud Diagram [1]

<sup>8</sup><https://www.w3.org/DesignIssues/LinkedData.html>



## 2.3 Interlinking

Interlinking is the process which connects and enhances data navigation and search between datasets through the Linked Data. The LOD principles core is to give data accessibility throughout the web supporting the data connectivity using RDF links.

E-learning is a relevant example of interlinking where the educational resources linked from different repositories which enable sharing of knowledge, interest, cultural, and technical environments of learners. [9]

### 2.3.1 Entity Linking

Entity linking or named entity normalization is the process to determine the identity of entities mentioned in a text concerning a knowledge base. In other words, grounds the entities to a knowledge base such as DBpedia and Wikipedia, or a relational database. Linked entities contribute to the successful entity detection boundaries and the correct classification of entity types, e.g., persons, locations, organizations, objects, and so on. Furthermore, extracting and recognizing data is a crucial process of the natural language processing field. [10]

### 2.3.2 Evaluation Measures for Interlinking

Evaluation measures are used to estimate the success level and the efficiency of retrieving relevant data on machine learning frameworks. In other words, to evaluate entity linking, we compare the system's predictions with the gold standard (the right labels of the items). There exist many evaluation measures. Figure 2.4 presents the four essential terms for evaluation measures [11]:

- **True positive (TP)**: refers to the number of predicted positives which were correct.
- **False positive (FP)**: refers to the number of predicted positives which were incorrect.
- **True negative (TN)**: refers to the number of predicted negatives which were correct.
- **False negative (FN)**: refers to the number of predicted negatives which were incorrect.

Actual Class	Predicted Class		
		Class = YES	Class = NO
	Class = YES	True Positive	False Negative
	Class = NO	False Positive	True Negative

FIGURE 2.4: Essential four terms for evaluation measures [2]

The next subsections focus on the accuracy evaluations as F1 score, precision, and recall.

### 2.3.2.1 Precision

The precision (P) denotes the proportion of Predicted Positive cases that are correctly Real Positives. [11]

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

### 2.3.2.2 Recall

The recall (R) is the proportion of Real Positive cases that are correctly Predicted Positive. [11]

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

### 2.3.2.3 F1 Score

The F1 score or F measure is the harmonic average of the precision and recall [12]. The range for F1 Score range is  $[0, 1]$ . The higher the F1 Score, the better is the accuracy.

$$F1 = \frac{2*P*R}{P+R}$$

## Chapter 3

# State of the Art

This chapter is divided into two sections: for the first section, we present an overview of existing works which solved the heterogeneous data sources problem in the Semantic Web. And the second section, we describe and compare Natural Language Processing for Named Entity Recognition tools and interlinking tools that later are used in the Automated Link Discovery process for the maritime domain.

### 3.1 Related Literature

There is significant work on heterogeneous data sources retrieval, entity linking and interlinking them with Linked Open Data in various domains. Much of the work employs Semantic Web technologies to facilitate retrieval and data linking. We describe several inspiring papers concerning entity linking and interlinking tasks.

Weichselbraun et al. [13] create a named entity linking component called Recognize that uses background knowledge obtained from LOD repositories and transforms heterogeneous data sources into a linked enterprise data repository such as DBpedia<sup>1</sup> and GeoNames<sup>2</sup>. The authors do not work with machine learning algorithms; instead, they work with pre-processing data modules. Namely, the Recognize process starts by querying LOD sources that later will be pre-processed identifying the information in three entities, name (person entity), structure (location entity), and context (organization entity) analyzer.

Raimond et al. [14] describe current efforts towards interlinking music-related datasets on the Web using two naïve approaches to tackling the resource matching problem and

---

<sup>1</sup><https://wiki.dbpedia.org/>

<sup>2</sup><https://www.geonames.org/>

identifying their failings. Then, the authors create an algorithm for interlinking dividing it into two contexts: the first one is to link a Creative Commons music dataset to an editorial and the second one to connect a personal music collection to corresponding web identifiers.

Saleem et al. [15] create an approach of RDFizing data from the Cancer Genome Atlas (TCGA) and linking its elements to the Linked Open Data (LOD) Cloud. The TCGA is a multidisciplinary pilot project to create a genetic mutations atlas responsible for cancer. To do that, the authors convert the data into RDF for later being linked with the LOD Cloud using LIMES interlinking framework.

Hassanzadeh et al. [16] develop the Linked Movie Database (LinkedMDB) project which uses a novel way of creating and managing large volumes of high-quality movie links by employing state-of-the-art approximate join techniques for linking and providing additional RDF metadata concerning the quality and the methods used for deriving them. LinkedMDB demonstrates the value of using link discovery and RDF publishing linkage metadata to promote dense RDF datasets interlinking.

Caraballo et al. [17] develop DRX tool which assists data publishers with the help of dataset interlinking with datasets collected from the LOD cloud. In other words, a data publisher may use DRX tool to identify datasets that potentially have resources to be interlinked given a specific dataset. DRX tool incorporates five main modules: the collection of data from datasets on the LOD, processing the data collected, grouping datasets applying cluster algorithms, providing dataset recommendations, and supporting dataset browsing.

Zhang et al. [18] create a semantic approach which allows users to extract, interlink and integrate geospatial data from various data sources using the Karma tool. The authors develop an easy way to obtain geospatial information from multiple data sources by using three steps, modeling, linking and integrating. In the modeling step, the data is converted in RDF, then stored them in the RDF repositories for later in the linking step, being linked with other repositories and last the integrating step which imports the new linking results in the repository and show the results with Karma tool to the final user.

Piedra et al. [19] design the SmartLand-LD framework to achieve data interoperability and integration by republishing data into linked data. SmartLand-LD is a flexible and distributed ecosystem of heterogeneous data sources which facilitate access and the mixture of a range of information. The process is simple SmartLand-LD will import new heterogeneous data sources, then the framework will extract, clean and transform the data and metadata into RDF. Next, using the interlinking, the SmartLand-LD will link

the existing data with other resources related. To later be queried with SPARQL and to help the information consumers understand their relation.

The Automated Link Discovery approach presented in this thesis uses the Named Entity Recognition and the interlinking tools for the extraction and linking of heterogeneous data sources in the maritime domain. In contrast to other methods which transform the data sources into Linked Open Data, the Automated Link Discovery will extract the metadata, identify entities, and interlink the entities with Linked Open Vocabularies.

## 3.2 Related Tools

### 3.2.1 Natural Language Processing for Named Entity Recognition

Natural Language Processing (NLP) is an area of research which examines how machines can understand and manipulate human (natural) language, text or speech [20]. NLP applications include some fields of studies, such as speech recognition, natural language understanding, artificial intelligence, named entity recognition and so on.

The master thesis focuses on the Natural Language Processing for Named Entity Recognition. Named Entity Recognition (NER) is a subtask of information extraction which attempts to understand and identify specific occurrences of words, as belonging named entity mentions (categories) in an unstructured or semi-structured text [21]. The categories are classified as person, organization, and location, as well as, temporal and numerical expressions.

In the last years, many studies have applied NLP to NER [22] to successfully advanced in state-of-the-art performances. Consequently, there are some reviews presented in surveys. Nadeau and Sekine [23] presented a survey applying NER in news articles and web pages evaluating a list of categorized features used in recognition and classification algorithms, e.g., supervised learning, semi-supervised learning, and others. Campos et al. [24] presented a survey of machine learning tools where described various fundamental steps for each selected tool. Furthermore, they evaluated the biomedical entities for each tool to allow to expose the current trends on machine-learning based solutions and compare the performance of each tool. Atdağ and Labatut [10] presented a comparison of NER tools for biographical texts. The study tried to solve the question, which tool is best for a specific situation. In this case, the study compared four NER tools and evaluated the accuracy and time performance of each tool. The best accuracy and performance results were obtained by the Stanford NER tool. Li et al. [25] compared the most representative methods for various state-of-the-art NER tools giving a best

comprehensive understanding. The survey focused more on NER in general domain and NER in English languages. At the same time, the paper presented the challenges of using NER tools, such as data annotations, informal text, and unseen entities.

This section presents an explanation and comparison between two NER tools, Stanford Named Entity Recognizer tool and Apache OpenNLP.

### 3.2.1.1 Stanford Named Entity Recognizer

Stanford Named Entity Recognizer<sup>3</sup> (Stanford NER) also called CRFClassifier is a Java tool for named entity recognition which provides a general implementation of the linear chain Conditional Random Field (CRF) sequence models. CRF represents the state-of-the-art in sequence modeling which represents the hidden state sequence probability of some observations [26]. To use the Stanford NER, the user needs to create the models on labeled data to build sequence models for NER.

A classifier model is a serialized model which contains a group of classes as people, organizations, and locations. Table 3.1 shows three classifiers that Stanford NER provides. These models use distributional similarity features, which provide considerable performance gain at the cost of increasing their size and runtime.

Classifier	Classes
english.all.3class	Location, person, organization
english.all.4class	Location, person, organization, misc
english.all.7class	Location, person, organization, money, percent, time, date

TABLE 3.1: Stanford Named Entity Recognizer classifiers

The Stanford NER tool can run in three modes of application, Stanford NER GUI application, single CRF NER Classifier from command-line, and programmatic use via API.

The **Stanford NER GUI application** provides a Java user interface application. Figure 3.1 presents a simple text example which has been classified using the Stanford NER GUI tool. The classifier used is *english.all.7class*. As seen in the figure, each class (entity tag) has a unique color representation, e.g., “Johny Deep” is a *PERSON*.

<sup>3</sup><https://nlp.stanford.edu/software/CRF-NER.shtml>

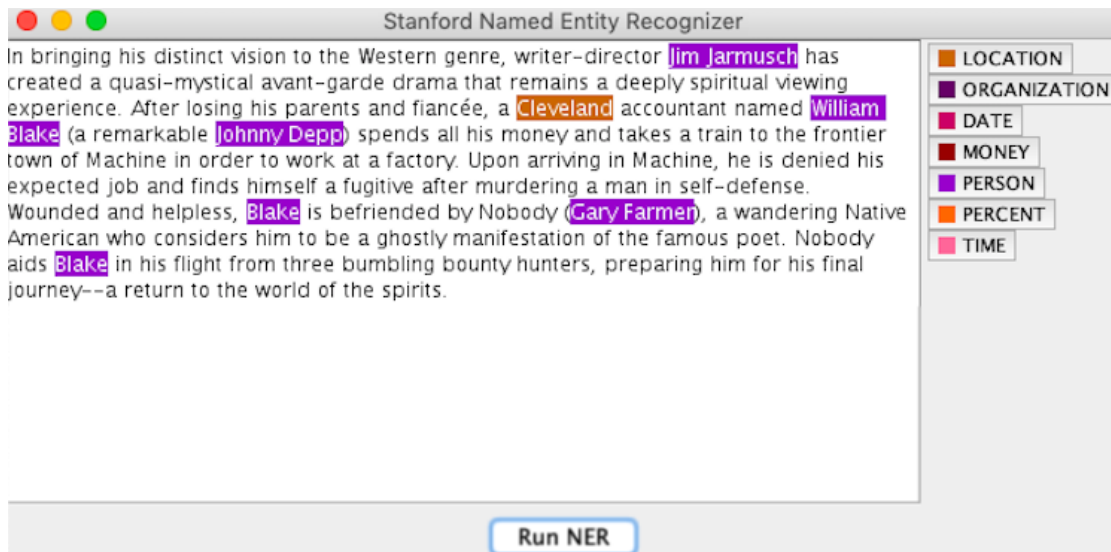


FIGURE 3.1: Stanford NER GUI tool

The main limitation of using this mode is that the Stanford NER GUI application does not permit the use of multiple classifier models. At the same time, the Stanford NER GUI application presented for demonstration and testing.

The **single CRF NER Classifier** is a command line tool where the user can interact with the tool in various styles. To use the single CRF NER Classifier the user needs a text file to be classified (sample.txt) and the classifier models. The following commands are for Unix/Linux machines.

Listing 3.1 presents the command line which prints the sentences with each word being classified by tags, e.g., *The/O fate/O of/O Lehman/ORGANIZATION Brothers/OR-  
GANIZATION...*

```
java -mx600m -cp "*/lib/*" edu.stanford.nlp.ie.crf.CRFClassifier -
  loadClassifier classifiers/english.all.3class.distsim.crf.ser.gz -
  textFile sample.txt
```

LISTING 3.1: Command line for printing named entities

Listing 3.2 presents the command line which returns the named entities classification in a TSV<sup>4</sup> file.

```
java -mx600m -cp "*/lib/*" edu.stanford.nlp.ie.crf.CRFClassifier -
  loadClassifier classifiers/english.all.3class.distsim.crf.ser.gz -
  outputFormat tabbedEntities -textFile sample.txt > sample.tsv
```

LISTING 3.2: Command line for printing named entities in a TSV file

<sup>4</sup>TSV stands for Tab Separated Values

Listing 3.3 is a command line that search for all the named entities using the three classifier models and then prints the results.

```
java -mx1g -cp "*/lib/*" edu.stanford.nlp.ie.NERClassifierCombiner -
  textFile sample.txt -ner.model classifiers/english.all.3class.
  distsim.crf.ser.gz, classifiers/english.conll.4class.distsim.crf.ser
  .gz, classifiers/english.muc.7class.distsim.crf.ser.gz
```

LISTING 3.3: Command line for printing named entities using the three classifier models

The **programmatic use via API** presents a set of functions allowing the creation of applications that access the features of the Stanford NER. Stanford NER presents a Java code which includes many ways to call the Stanford NER tool programmatically, see listing 3.4.

```
public class NERDemo {
    public static void main(String[] args) throws Exception {
        String serializedClassifier = "classifiers/english.all.3class.
        distsim.crf.ser.gz";
        if (args.length > 0) {
            serializedClassifier = args[0];
        }
        AbstractSequenceClassifier<CoreLabel> classifier = CRFClassifier.
        getClassifier(serializedClassifier);
        /* For either a file to annotate or for the hardcoded text
        example, this demo file shows several ways to process the input,
        for teaching purposes.
        */
        if (args.length > 1) {
            /* For the file, it shows (1) how to run NER on a String, (2)
            how to get the entities in the String with character offsets, and
            (3) how to run NER on a whole file (without loading it into a
            String).
            */
            String fileContents = IOUtils.slurpFile(args[1]);
            List<List<CoreLabel>> out = classifier.classify(fileContents);
            for (List<CoreLabel> sentence : out) {
                for (CoreLabel word : sentence) {
                    System.out.print(word.word() + '/' + word.get(
                    CoreAnnotations.AnswerAnnotation.class) + ' ');
                }
                System.out.println();
            }
            System.out.println("——");
            out = classifier.classifyFile(args[1]);
```



```

    for (List<CoreLabel> sentence : out) {
        for (CoreLabel word : sentence) {
            System.out.print(word.word() + '/' + word.get(
CoreAnnotations.AnswerAnnotation.class) + ' ');
        }
        System.out.println();
    }
    ...

```

LISTING 3.4: A snippet code from the API code presented by Stanford NER

### 3.2.1.2 Apache OpenNLP

Apache OpenNLP<sup>5</sup> is an open source machine learning library for processing natural language text. Apache OpenNLP supports the most common applications in NLP, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction and so on.

Apache OpenNLP for named entity recognition distributes a pre-built collection of models which allow the developer to identify proper nouns and numeric amounts to tag them into categories semantically. OpenNLP has seven categories, people, location, organization, date, time, percentage, and money. Each category has its model [27]. Therefore the developer needs to specify the model which is going to work.

The Apache OpenNLP tool can run two modes of applications, a command-line interface (CLI), and via an application program interface (API).

The **Command-line interface** provides a better convenience of experiments and training with the tool. However, CLI is only intended for demonstration and testing. For Apache OpenNLP, the CLI application has six types of methods, such as `TokenNameFinder`, `TokenNameFinderTrainer`, `TokenNameFinderEvaluator`, `TokenNameFinderCrossValidator`, `TokenNameFinderConverter`, and `CensusDictionaryCreator`. For example listing 3.5 presents the CLI method *TokenNameFinder* which returns the named entities given a text and model(s).

```

opennlp TokenNameFinder model1 model2 ... modelN < sentences

```

LISTING 3.5: Command line for printing named entities using Apache OpenNLP

On the case of listing 3.6, presents an example of executing the model *en-ner-person.bin* which it will only returns tags of *person* in English.

<sup>5</sup><https://opennlp.apache.org/>

```
opennlp TokenNameFinder en-ner-person.bin < Pierre Vinken , 61 years
old , will join the board as a nonexecutive director Nov. 29 .
```

LISTING 3.6: Example of Command line for printing named entities using Apache OpenNLP

```
<START:person> Pierre Vinken <END> , 61 years old , will join the
board as a nonexecutive director Nov. 29 .
```

LISTING 3.7: Output of the listing 3.6

The **application program interface** is strongly recommended to embed the OpenNLP features directly into the application. To use the API, first, load the model into memory to avoid common errors when executing the application. Listing 3.8 presents a snippet code example of using the Apache OpenNLP.

```
// Load the model en-ner-person.bin in memory
try (InputStream modelIn = new FileInputStream("en-ner-person.bin")) {
    TokenNameFinderModel model = new TokenNameFinderModel(modelIn);
}
// Instantiation of NameFinderME
NameFinderME nameFinder = new NameFinderME(model);
String sentence [] = new String [] {
    "Pierre",
    "Vinken",
    "is",
    "61",
    "years",
    "old",
    "."
};
Span nameSpans [] = nameFinder.find(sentence);
```

LISTING 3.8: A snippet of the API code

### 3.2.1.3 Comparison between Stanford NER and Apache OpenNLP

In this section, a comparison between the two natural language processing tools (Stanford NER and Apache OpenNLP) was performed to show which one is suitable to implement in this master thesis. Table 3.2 presents a summary comparison between both tools.

	<b>Stanford NER</b>	<b>Apache OpenNLP</b>
Benefits	Stanford NER detects all tags with a single annotator; is faster and has better accuracy detection than other known NLP tools; is a benchmark/state-of-the-art for NLP; extracts all tags with only one model	Apache OpenNLP training models are easier-to-use for not off-the-shelf training; has a good API for integration
Limitations	Poor documentation for training models	Apache OpenNLP is not a state-of-the-art for NLP; OpenNLP has a trained model for each specific tag i.e., it has a model for POS tags, location NER tags, person NER tags
Programming language	Java	Java
Training difficulty	Easy (supervised training data)	Easy (unsupervised training data)
Number of available models	3 (only english) and 6 in other languages (e.g., Spanish, Chinese) [28]	> 20
License	Open source	Open source
GUI application	Yes (Java application)	No
RESTful API	Yes	Yes
CLI application	Yes	Yes
Version	3.9.2	1.9.1

TABLE 3.2: Comparison between Stanford NER and Apache OpenNLP

Stanford NER is a tool which provides an easy way of using many types of features for extracting the named entities from text. Similarly, the OpenNLP tool performs easier-to-use tasks for named entity recognition. Furthermore, both tools require tokenization before doing any extraction. Though Stanford NER can extract all tags specified by a single annotator, Apache OpenNLP has a trained model for each specific tag. In other words, when a developer wants to extract the person, organization and location tags using Stanford NER, the developer only needs to use the model *english.all.3class*. While Apache OpenNLP, the developer needs to use three models each for the tags categories.

Concerning state-of-the-art technology approach for NLP between both tools, only Stanford NER applies for robust, broad-coverage natural-language processing in many languages. Stanford NER accuracy and time performance are better than any other existing named entity recognition tool.

### 3.2.2 Interlinking Tools

The interlinking tools are an easy and fast way of connecting data items to the Linked Open Data clouds [9]. Interlinking tools identify similarities between entities and create links between the source and the target entities, e.g., owl:sameAs. Nonetheless, the interlinking tools use various approaches which offer different functionalities and services.

Numerous studies have undertaken to evaluate the differences between the interlinking tools which affect three essential performance evaluations such as the performance time, accuracy performance, and resources used in memory and CPU.

Wölger et al. [29] presented a study where they compare with important aspects and describe (general and technical aspects) the interlinking tools available. Furthermore, they added additional information regarding the interlinking techniques for completeness. Another similar study was presented by Ferrara et al. [30], in this case, the study concentrated on the underlying techniques of the interlinking tools and how the interlinking tools behave when there are problems related to the entity resolution or object matching.

On the other hand, the next studies focused on comparing the currently available interlinking tools which could be applied to solve specific linking tasks. Simperl et al. [31] proposed a study where they compare interlinking tools by discussing the critical aspect such as input, output, the considered domain, and the linking techniques which rely on human contributions. Rajabi et al. [9] consider various interlinking tools in which they evaluated using a set of predefined criteria. The evaluation was carried out over the target datasets (e-learning repositories), and the linked results were presented to experts for validation; only two tools have more promising results concerning how the data can rely on current interlinking tools and thus adopt them to connect their resources to the Linked Open Data, namely Silk and LIMES. Nentwig et al. [32] presented a survey of current Link Discovery frameworks, eleven in total, in which were compared based on a standard set of criteria. The criteria cover the configuration for linking the data sources, methods for linking, runtime optimizations, and many more.

In this section, a structured explanation and comparison between LIMES and Silk tools are presented.

### 3.2.2.1 LIMES

LIMES<sup>6</sup> stands for Link Discovery Framework for Metric Spaces. LIMES implements mathematical characteristics of metric spaces to discover interlinks between data sources as well as machine learning algorithms to (semi-) automatically learn link specifications [33].

LIMES computes deterministic approximations of the similarity of instances to filter a large amount of data/instances which do not satisfy the linking conditions. At the same time, LIMES will save the instances extracted from the data sources in the cache to provide better efficiency and performance of the tool.

LIMES can run in two modes, CLI<sup>7</sup> client/server, and GUI<sup>8</sup> client. All two services offer the same functionalities. However, each of them works with a different interface to support different use cases.

Figure 3.2 presents LIMES GUI when it has started for the first time. The primary purpose of LIMES GUI is to provide the users with an easy way to configure LIMES without the need to code LIMES configuration in XML or RDF format. LIMES GUI has three components, the menu bar where the user can set up a new configuration between two datasets endpoint, the toolbox where the user can select the properties, measures, and operators of the data sources to link, and the metric builder where the user inserts the process of link specification creation.

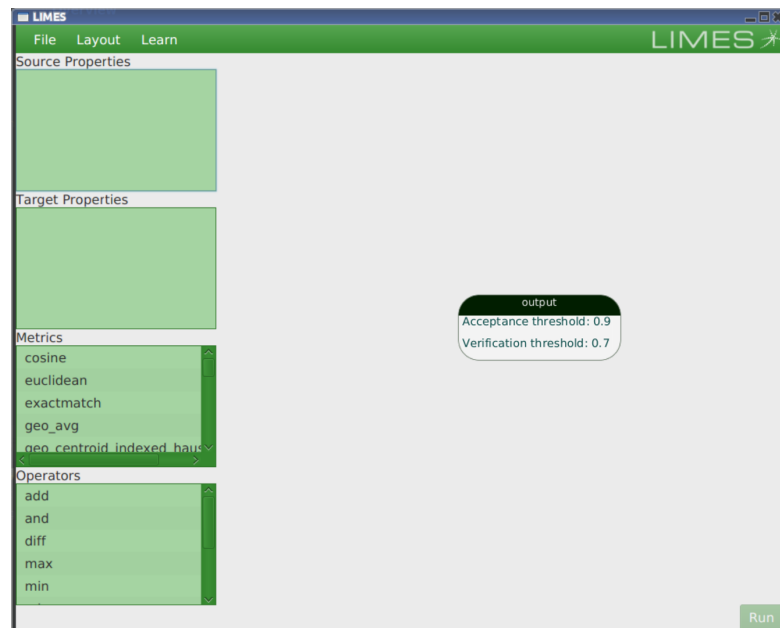


FIGURE 3.2: LIMES GUI

<sup>6</sup><http://aksw.org/Projects/LIMES.html>

<sup>7</sup>CLI stands for Command Language Interpreter

<sup>8</sup>GUI stands for Graphical User Interface

In the case of CLI client/server, LIMES can run as a RESTful API, the configuration files are accepted via POST multipart/form-data uploads. Each configuration execution returns a unique id to use later for querying the status, the logs, and the list of results files obtained by LIMES. After writing the configuration file, LIMES can be executed from the CLI client by calling *java -jar limes.jar config.xml -s*.

As explained before, to use LIMES, it is needed a configuration file which can be written in XML or RDF serialization. The next sections explained the five parts of LIMES configuration in an XML format such as metadata, prefixes, source/target data sources, metrics for similarity measurement, acceptance/review conditions, and output format.

#### 3.2.2.1.1 Metadata

The metadata always consists of a reference to an external DTD<sup>9</sup> file, see listings 3.9.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE LIMES SYSTEM "limes.dtd">
<LIMES>
```

LISTING 3.9: Metadata of LIMES configuration file

#### 3.2.2.1.2 Prefixes

The prefixes are used in LIMES to identify the namespace which will be addressed by the prefix's label. In other words, the prefixes contain two parts: the namespace, and the label. The namespace stores the URI, and the label is the short name of the namespace. LIMES configuration file allows as many prefixes as needed in the configuration file, see listing 3.10.

```
<PREFIX>
<NAMESPACE>http://www.w3.org/1999/02/22-rdf-syntax-ns#</NAMESPACE>
<LABEL>rdf</LABEL>
</PREFIX>
```

LISTING 3.10: Prefix of LIMES configuration file

#### 3.2.2.1.3 Data Sources

The data sources are two Linked Data sources called source and target. Both sources need to be configured using the same tags. The *ID* is a unique name to identify each

<sup>9</sup>DTD stands for document type definition.

of the sources. The *ENDPOINT* specifies the path of the file containing the data or SPARQL endpoint for the data source. The *VAR* is the variable associated with the endpoint that later will be used by the metric expression. *PAGESIZE* is the maximal number of triples which can be returned by the SPARQL endpoint. *RESTRICTION* limits the query results on the SPARQL endpoint. *PROPERTY* is used to pre-process the input data and to specify the property that will use in metric comparison. *TYPE* defines the type of endpoint, like CSV, N3<sup>10</sup>, TURTLE, and SPARQL. Listings 3.11 presents an example of data sources.

```
<SOURCE>
  <ID>mesh</ID>
  <ENDPOINT>http://mesh.bio2rdf.org/sparql</ENDPOINT>
  <VAR>?y</VAR>
  <PAGESIZE>5000</PAGESIZE>
  <RESTRICTION>?y rdf:type meshr:Concept</RESTRICTION>
  <PROPERTY>dc:title</PROPERTY>
  <TYPE>sparql</TYPE>
</SOURCE>
<TARGET>
  <ID>linkedct</ID>
  <ENDPOINT>http://data.linkedct.org/sparql</ENDPOINT>
  <VAR>?x</VAR>
  <PAGESIZE>5000</PAGESIZE>
  <RESTRICTION>?x rdf:type linkedct:condition</RESTRICTION>
  <PROPERTY>linkedct:condition_name</PROPERTY>
</TARGET>
```

LISTING 3.11: Data Sources of LINES configuration file

#### 3.2.2.1.4 Metrics

The metrics are the metric expression for similarity measurement which specifies the selected metric and the operations to calculate an estimation value of similarity between instances in the source and the target data sources. Table 3.3 shows the different packages of linking measures which LINES supports. In total are six packages, each of them operates a specific type of resource. Type of resource as strings, numbers, vectors, polygons, time, geometric forms and even set of RDF.

<sup>10</sup><https://www.w3.org/TeamSubmission/n3/>

Linking Measure Packages	Definition
String Measure	Metric which measures the distances between two strings.
Vector Space Measure	Comparison between numeric vectors by using the vector space measures.
Point-set Measure	The similarity between polygons can be measured using the point-set distance.
Topological Measure	The topological measure is the relation between the spatial representation of points of interest (POI) resources. LIMES assumes the link between different POI has a geospatial representation in the form of geometry.
Temporal Measure	The temporal measure is the relation between the temporal representation of POI resources. LIMES support temporal measures based on Allens algebra.
Resource-set Measure	Comparison of sets of resources in RDF containers using Jaccard.

TABLE 3.3: LIMES Measure packages.

For this master thesis, we will only use the string measures package. LIMES supports twelve types of string matching [34]:

- **Cosine:** Cosine metric is a measure of similarity between two vectors of an inner product space which measures the cosine of the angle between them. The cosine of  $0^\circ = 1$  and it is less than 1 for any other angle [35].

$$Similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

- **Exact Match:** Exact Match string similarity is a measure of similarity between two strings which returns 1 if they are identical, 0 otherwise.

- **Jaccard:** The Jaccard index or Jaccard similarity coefficient compares members to find the similarity and diversity of sample sets.

$$Jaccard = J(X, Y) = \frac{X \cap Y}{X \cup Y}$$

- **Jaro:** The Jaro distance is a measure of similarity between two strings which are related to the number of single-character transpositions required to change the order of one word into the other.



- **Jaro-Winkler:** The Jaro-Winkler distance is a measure of similarity between two strings which use a prefix scale that gives more favorable ratings. The lower the distance is, the more similar the strings are.

$$s = \frac{m}{3*a} + \frac{m}{3*b} + \frac{m-t}{3*m}$$

where  $m$  is the number of matching characters,  $t$  is half the number of transpositions,  $a$  is the length of the first string, and  $b$  is the length of the second string.

- **Levenshtein:** Levenshtein distance is a metric for measuring the amount of difference between two sequences. It defined as the minimum number of edits needed to transform one string into another, with the allowable edit operations insert, substitute, or delete a single character.
- **Trigram:** Trigram is a group of three consecutive characters taken from a string. Trigram can measure the similarity of two strings by counting the number of trigrams they share.
- **Qgrams:** The Qgrams measure is similar to trigram measure. Qgrams uses a group of  $q$  consecutive characters for generating the vectors of the string.
- **Soundex:** Soundex has the property of indexing names by sound in English. The goal is to form words that produce the same sound key and thus be used to simplify searches in databases where it is known the pronunciation but not the spelling. In LIMES, we compute the Soundex distance as the reverse of the distance between the encoding of the two input strings.
- **Overlap:** Overlap measures the overlap between two sets. It is related to the Jaccard index and defined as the size of the intersection divided by the smaller of the size of the two sets.
- **Monge Elkan:** It is a type of hybrid similarity measure that combines the benefits of sequence-based and set-based methods. This measure is useful for domains in which more control needed over the similarity measure.
- **Ratcliff Obershelp:** The Ratcliff Obershelp measures the similarity of two strings as the number of matching characters in the two strings.

$$D_{ro} = \frac{2*km}{|S_1|+|S_2|}$$

where  $km$  is the number of matching characters,  $S$  is the string length.

Listing 3.12 presents an example of metrics, in this case, trigrams similarity function used to identify possible links between the title of the source and the condition name of the target.

```
<METRIC>
  trigrams(y.dc:title , x.linkedct:condition_name)
</METRIC>
```

LISTING 3.12: Metrics of LINES configuration file

### 3.2.2.1.5 Conditions

The conditions are divided into two types called acceptance and review. The acceptance condition is the satisfied links between two instances, while the review condition is the links which could be satisfied and therefore needs to verify manually. Both conditions need to be configured using the same tags. The *THRESHOLD* is the minimum value that two instances must have in order to satisfy. The *FILE* is the name of the file where the linked results are going to be stored. The *RELATION* is the relation between the instances of each data source, i.e., owl:sameAs.

Listing 3.13 presents an example of the conditions. As seen in the listing the acceptance will save the links which threshold is between 0.98 and 1.0. While the review condition threshold is between 0.95 and 0.98.

```
<ACCEPTANCE>
  <THRESHOLD>0.98</THRESHOLD>
  <FILE>accepted.nt</FILE>
  <RELATION>owl:sameAs</RELATION>
</ACCEPTANCE>
<REVIEW>
  <THRESHOLD>0.95</THRESHOLD>
  <FILE>reviewme.nt</FILE>
  <RELATION>owl:sameAs</RELATION>
</REVIEW>
```

LISTING 3.13: Conditions of LINES configuration file

### 3.2.2.1.6 Output

The output is the tag for specifying the format of the output file. LINES only supports output in N3 or TAB, see listing 3.14.

```
<OUTPUT>N3</OUTPUT>
```

LISTING 3.14: Output tag of LINES configuration file

At the same time, LINES has optional tags for the configuration file as execution, machine learning, and granularity. Those optional tags are for interlinking data sources with other types of complex configurations and methods.

### 3.2.2.2 Silk

Silk Link Discovery Framework<sup>11</sup> is an open source tool for interlinking data items within different Linked Data sources with the possibility to apply data transformation to structured data sources on the web. Using Silk - Link Specification Language (Silk-LSL), the users can specify the types of RDF links between two data sources, as well as, the conditions which need to fulfill to interlink, like, similarity metrics, comparison, aggregator, and transformation functions [36], see Figure 3.3 for an example of Silk-LSL.

```

<Silk>
  <Prefixes>
    <Prefix id="rdf" namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
    <Prefix id="dbpp" namespace="http://dbpedia.org/property/" />
    <Prefix id="dcterms" namespace="http://purl.org/dc/terms/" />
    <Prefix id="dc" namespace="http://purl.org/dc/elements/1.1/" />
    <Prefix id="owl" namespace="http://www.w3.org/2002/07/owl#" />
    <Prefix id="foaf" namespace="http://xmlns.com/foaf/0.1/" />
    <Prefix id="rdfs" namespace="http://www.w3.org/2000/01/rdf-schema#" />
    <Prefix id="dbpediaowl" namespace="http://dbpedia.org/ontology/" />
    <Prefix id="linkedmdb" namespace="http://data.linkedmdb.org/resource/movie/" />
  </Prefixes>
  <DataSources>
    <Dataset id="DBpedia" type="file">
      <Param name="file" value="source.nt" />
      <Param name="format" value="N-TRIPLE" />
      <Param name="graph" value="" />
    </Dataset>
    <Dataset id="linkedmdb" type="file">
      <Param name="file" value="target.nt" />
      <Param name="format" value="N-TRIPLE" />
      <Param name="graph" value="" />
    </Dataset>
  </DataSources>
  <Interlinks>
    <Interlink id="movies">
      <SourceDataset dataSource="DBpedia" var="a" typeUri="http://dbpedia.org/ontology/Film">
        <RestrictTo> </RestrictTo>
      </SourceDataset>
      <TargetDataset dataSource="linkedmdb" var="b" typeUri="http://data.linkedmdb.org/resource/movie/film">
        <RestrictTo> </RestrictTo>
      </TargetDataset>
      <LinkageRule linkType="owl:sameAs">
        <Aggregate id="combineSimilarities" required="false" weight="1" type="min">
          <Compare id="compareTitles" required="false" weight="1" metric="levenshteinDistance" threshold="0.0" indexing="true">
            <TransformInput id="toLowerCase1" function="toLowerCase">
              <Input id="movieTitle1" path="/foaf:name" />
            </TransformInput>
            <TransformInput id="toLowerCase2" function="toLowerCase">
              <Input id="movieTitle2" path="/rdfs:label" />
            </TransformInput>
            <Param name="minChar" value="0" />
            <Param name="maxChar" value="z" />
          </Compare>
          <Compare id="compareReleaseDates" required="false" weight="1" metric="date" threshold="400.0" indexing="true">
            <Input id="movieReleaseDate1" path="/dbpediaowl:releaseDate" />
            <Input id="movieReleaseDate2" path="/linkedmdb:initial_release_date" />
          </Compare>
        </Aggregate>
        <Filter />
      </LinkageRule>
      <Outputs> </Outputs>
    </Interlink>
  </Interlinks>
</Transforms> </Transforms>
</Outputs> </Outputs>
</Silk>

```

FIGURE 3.3: Example of Silk-LSL

<sup>11</sup><http://silkframework.org/>

Silk provides **Silk Workbench** a web application where the users can process the interlinking of different data sources with the help of a user interface. Silk Workbench contains three components, workspace browser, linkage rule editor, and evaluation [37].

The **Workspace browser** permits users to manage their projects, where a project holds the data sources and linking tasks defined. An example of the workspace browser displayed in figure 3.4 where in this case there is a project called movies, and it has two N-Triple data sources with a linking called involves.

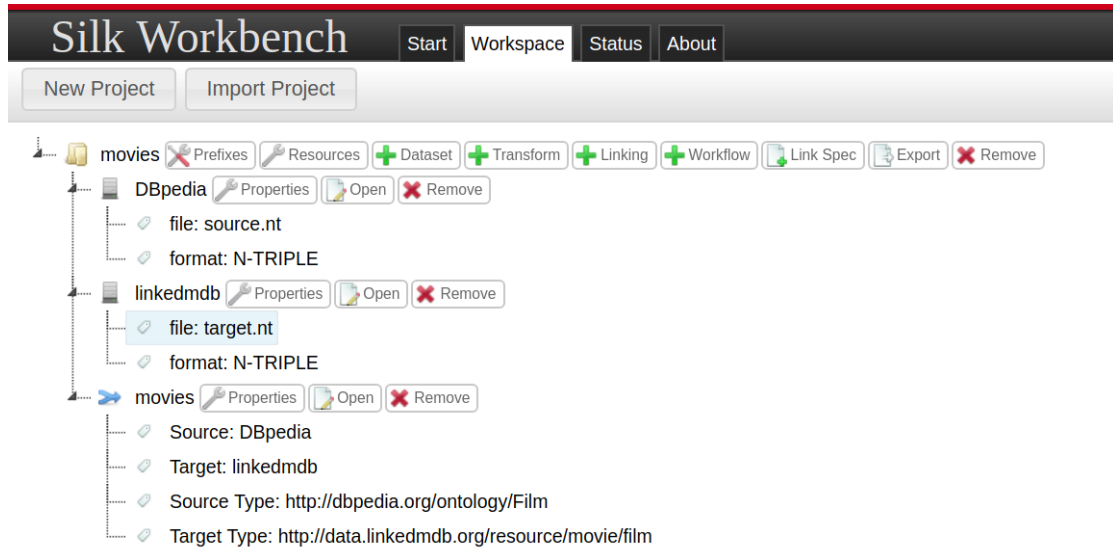


FIGURE 3.4: Workspace browser component of Silk Workbench

The **linkage rule** is a graphical editor which allows users to manage the links between the two data sources. The linkage rules are created as an operator tree by dragging and dropping the rule elements. Figure 3.5 presents a linkage rule example. As shown in the figure, the linkage rule has two panes; the left pane contains the property paths for the source and the target data sources, and the restrictions as the transformations, comparators (see table 3.4), and aggregators which can be used to transform the input data. The right pane is where the elements are dropped to create the operator tree which describes the data sources interlinking.

The **evaluation** consists of the results generated by the linkage rule of a project. Figure 3.6 presents an example of results generated by Silk when executing the tool. In the evaluation, each link can be classified as correct or incorrect. The process is called correctness, and it needs to be classified manually when there is no classification. The purpose of the correctness is to measure the success of the interlinked entity and to find potential errors.

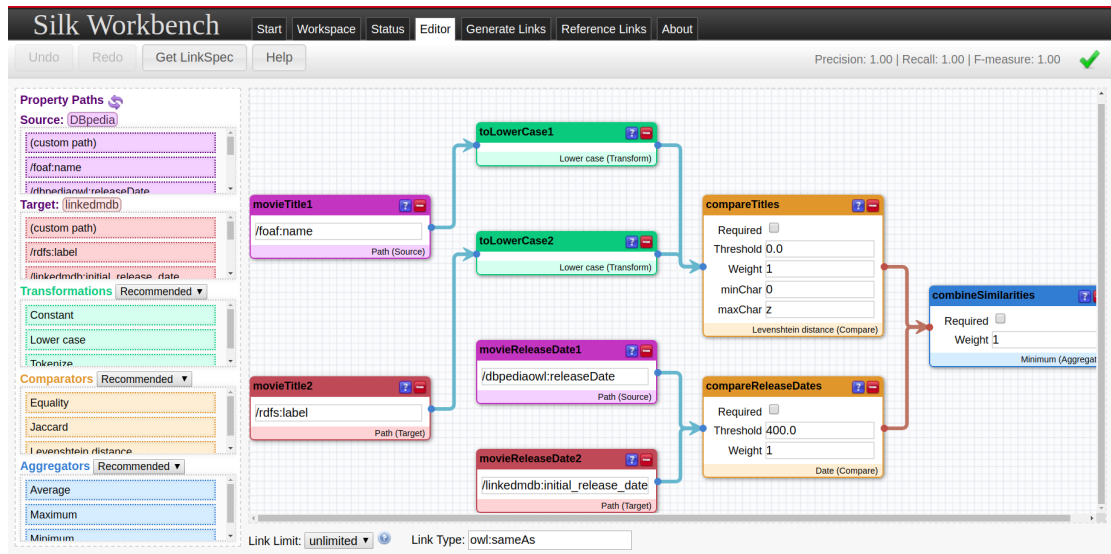


FIGURE 3.5: Linkage rule component of Silk Workbench

Comparators	Similarity Measures
Character based	Jaro distance, Jaro-Winkler distance, Levenshtein distance, SubString, Qgrams
Numeric	Date, Date Time, Geographical distance, Numeric similarity, ...
Spatial	Centroid distance, Crosses, Disjoint, Intersects, Min distance, ...
Temporal	Hours distance, Days distance, Minutes distance, ...
Token based	Cosine, Dice coefficient, Jaccard, Soft Jaccard, Token-wise distance

TABLE 3.4: Silk measure packages

Source: DBpedia	Target: linkedmdb	Score	Correct
http://dbpedia.org/resource/Jagged_Edge_%28film%29	http://data.linkedmdb.org/resource/film/2304	100.0%	✓ ? ✗
http://dbpedia.org/resource/Where_Are_My_Children%3F	http://data.linkedmdb.org/resource/film/236	100.0%	✓ ? ✗
http://dbpedia.org/resource/Foxy_Brown_%28film%29	http://data.linkedmdb.org/resource/film/2519	99.0%	✓ ? ✗
http://dbpedia.org/resource/Tillie%27s_Punctured_Romance_%281914_film%29	http://data.linkedmdb.org/resource/film/600	100.0%	✓ ? ✗
http://dbpedia.org/resource/Tora%21_Tora%21_Tora%21	http://data.linkedmdb.org/resource/film/82	100.0%	✓ ? ✗
http://dbpedia.org/resource/Toys_%28film%29	http://data.linkedmdb.org/resource/film/999	100.0%	✓ ? ✗
http://dbpedia.org/resource/Little_Nicky	http://data.linkedmdb.org/resource/film/1749	97.5%	✓ ? ✗
http://dbpedia.org/resource/Hard_Eight_%28film%29	http://data.linkedmdb.org/resource/film/1184	100.0%	✓ ? ✗
http://dbpedia.org/resource/Jesus_%281979_film%29	http://data.linkedmdb.org/resource/film/682	95.5%	✓ ? ✗
http://dbpedia.org/resource/Castaway_%28film%29	http://data.linkedmdb.org/resource/film/2051	100.0%	✓ ? ✗
http://dbpedia.org/resource/Cape_Fear_%281991_film%29	http://data.linkedmdb.org/resource/film/442	100.0%	✓ ? ✗
http://dbpedia.org/resource/Twisted_%282004_film%29	http://data.linkedmdb.org/resource/film/1684	100.0%	✓ ? ✗
http://dbpedia.org/resource/Underworld:_Evolution	http://data.linkedmdb.org/resource/film/2451	99.8%	✓ ? ✗
http://dbpedia.org/resource/Vanity_Fair_%282004_film%29	http://data.linkedmdb.org/resource/film/2181	100.0%	✓ ? ✗
http://dbpedia.org/resource/Vanishing_Point_%281971_film%29	http://data.linkedmdb.org/resource/film/2282	85.8%	✓ ? ✗
http://dbpedia.org/resource/Garden_State_%28film%29	http://data.linkedmdb.org/resource/film/1082	100.0%	✓ ? ✗
http://dbpedia.org/resource/National_Treasure_%28film%29	http://data.linkedmdb.org/resource/film/1298	100.0%	✓ ? ✗

FIGURE 3.6: Evaluation component of Silk Workbench

### 3.2.2.3 Comparison between LIMES and Silk

In this section, a comparison between the two interlinking tools (LIMES and Silk) was performed to show which one is suitable to implement in this project. Table 3.5 presents a summary comparison between both tools.

	<b>LIMES</b>	<b>Silk</b>
Benefits	LIMES implements time efficiency approaches for interlinking; implements machine learning EAGLE and WOMBAT algorithms; It is 60 times faster; detects "duplicates" in a single source	Silk has scalable and high performance through efficient data handling; Silk Workbench presents a straightforward graphical user interface for interlinking
Limitations	LIMES has insufficient memory problems	The Silk Workbench evaluation component of the generated links must implement manually
Type of data sources	SPARQL endpoints, CSV, and RDF files	SPARQL endpoints, CSV, XML, RDF files
Functions	Similarity measures, transformation, aggregation functions, and boolean operations	Comparator, transformation, and aggregation functions
License	Open source	Open source
GUI application	Yes (Java application)	Yes (web application)
RESTful API	Yes	No
Version	1.5.0 in 2018-08-20	2.7.1 in 2016-02-24

TABLE 3.5: Comparison between Limes and Silk

LIMES is a framework which provides the necessary functionalities to discover missing interlinks between Link Data sources. At the same time, LIMES encapsulates complex algorithms dedicated to the processing of structured data of any sort, like similarity measurements, and machine learning algorithms (EAGLE and WOMBAT). LIMES reduces the number of comparisons needed during mapping using the caching method. Additionally, LIMES detects duplicates in one or both data sources by using the string metrics and uses the triangle inequality to portion the metric space [33]. Therefore,

this results in a performance boost allowing LIMES to execute faster than other existing interlinking tools and improves the performance of continuous queries. However, depending on how significant is the data source, this can affect the available memory.

On the other hand, Silk divides the main tasks into smaller tasks, helping to execute only subtasks in a multi-thread method. In this case, it is less likely to encounter insufficient memory problems when using Silk. Nevertheless, Silk only works as a user interface web application, while LIMES permits the use of the features in the GUI java application and the client/server application (RESTful API).

## Chapter 4

# BigDataOcean Project

BigDataOcean (BDO) is a European Project that aims to enable maritime big data scenarios for EU-based companies, organizations, and scientists, through a multi-segment platform that will combine data of different velocity, variety, and volume under an inter-linked trusted multilingual engine. [4]

The BDO project has four pilot cases [4],

1. **Fault Prediction and Proactive Maintenance** aims to provide predictions about damages, unpredicted damages and mechanical failures of vessels, and environmental damages caused.
2. **Mare Protection** requires the description of atmospheric, wave and hydrodynamical data, combined with location, rate, and characteristics of an oil spill to increase the forecasting capabilities,
3. **Maritime Security and Anomaly Detection**, need data about security, events, and threats in the maritime environment to identify patterns that will impact security, economy, and environment, such as terrorism, illegal trafficking, and fishing,
4. **Wave Power** aims to contribute to the wave energy industry by offering the ability to predict the best locations, the expected energy production, equipment costs, and environmental impact. For this reason, storing environmental and geophysical data coming from vessels, buoys are essential.

The next sections, we only concentrate on two of the tools which BigDataOcean offers, BigDataOcean Vocabulary Repository and BigDataOcean Harmonization Tool.



## 4.1 BigDataOcean Vocabulary Repository

BigDataOcean Vocabulary Repository forms part of the infrastructure to access and share Linked Big Data vocabularies and ontologies related to the maritime domain. Figure 4.1 shows the variety of vocabularies currently indexed in the BDO Vocabulary Repository [38].

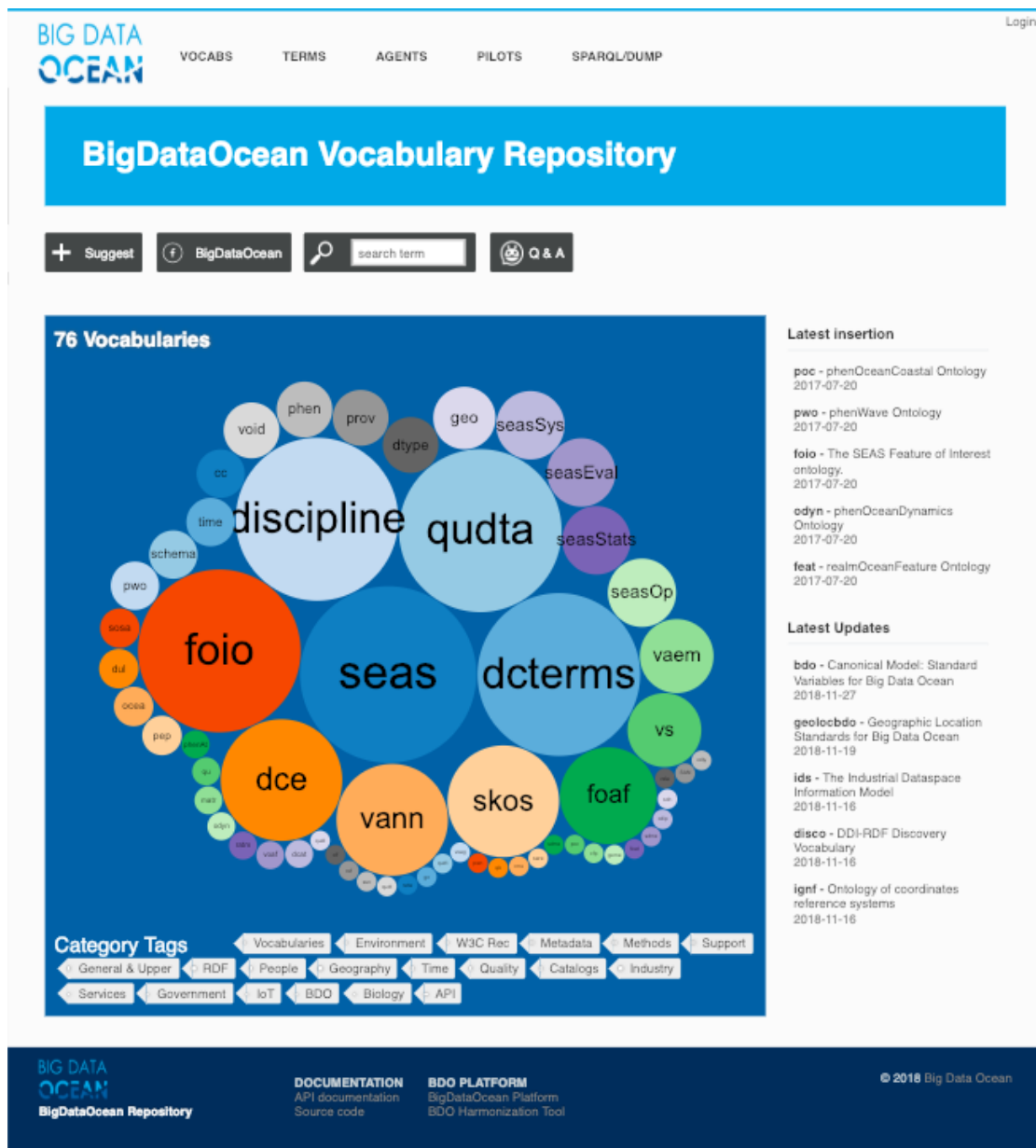
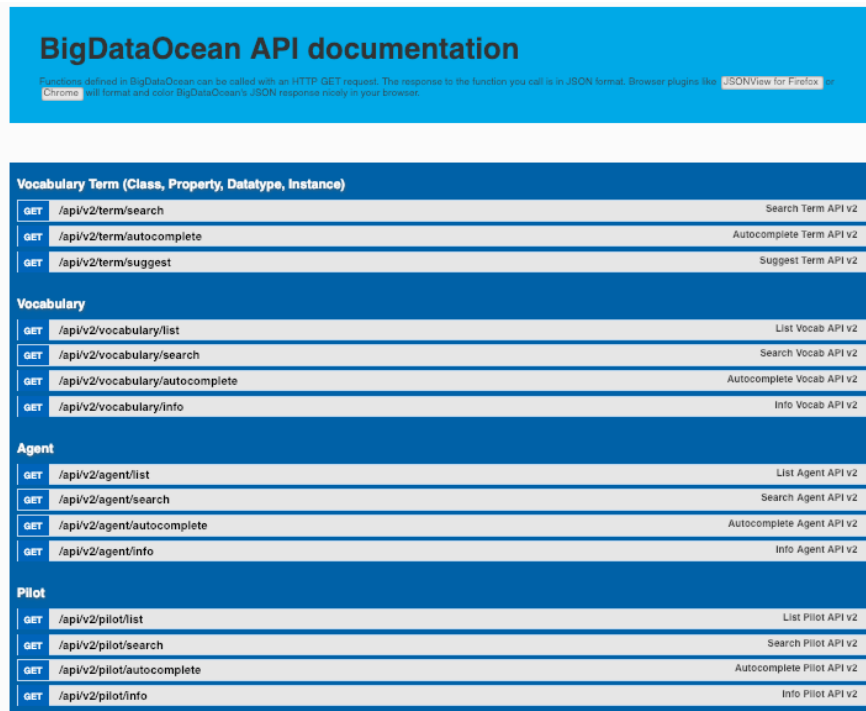


FIGURE 4.1: BigDataOcean Vocabulary Repository.

The BDO Vocabulary Repository offers functions over HTTP GET requests for the different type of information as vocabularies, terms of specific vocabulary, agents and pilots. Figure 4.2 presents the APIs that Vocabulary Repository offers. By clicking

on each of the boxes, the BDO Vocabulary Repository presents which parameters are necessary to execute each API correctly.



**BigDataOcean API documentation**

Functions defined in BigDataOcean can be called with an HTTP GET request. The response to the function you call is in JSON format. Browser plugins like [JSONView for Firefox](#) or [Chrome](#) will format and color BigDataOcean's JSON response nicely in your browser.

Vocabulary Term (Class, Property, Datatype, Instance)		
GET	/api/v2/term/search	Search Term API v2
GET	/api/v2/term/autocomplete	Autocomplete Term API v2
GET	/api/v2/term/suggest	Suggest Term API v2
Vocabulary		
GET	/api/v2/vocabulary/list	List Vocab API v2
GET	/api/v2/vocabulary/search	Search Vocab API v2
GET	/api/v2/vocabulary/autocomplete	Autocomplete Vocab API v2
GET	/api/v2/vocabulary/info	Info Vocab API v2
Agent		
GET	/api/v2/agent/list	List Agent API v2
GET	/api/v2/agent/search	Search Agent API v2
GET	/api/v2/agent/autocomplete	Autocomplete Agent API v2
GET	/api/v2/agent/info	Info Agent API v2
Pilot		
GET	/api/v2/pilot/list	List Pilot API v2
GET	/api/v2/pilot/search	Search Pilot API v2
GET	/api/v2/pilot/autocomplete	Autocomplete Pilot API v2
GET	/api/v2/pilot/info	Info Pilot API v2

FIGURE 4.2: List of APIs which BigDataOcean Vocabulary Repository offers.

To have access to the RDF file of each vocabulary, the BDO Vocabulary Repository offers an API that gives the user, the metadata information of the vocabulary. For instance, figure 4.3 presents the vocabulary metadata with prefix *bdo* using the URL <http://localhost:3333/dataset/bdo/api/v2/vocabulary/info?vocab=bdo>.

We have created two ontologies particularly for the BDO project,

- **Geographic Location Standards for BigDataOcean:** A vocabulary which represents the geographic location standards used in the BigDataOcean project which representS the marine georeferenced place names and areas.<sup>1</sup>
- **Canonical Model Standard Variables for BigDataOcean:** A vocabulary which represents the variables defined in datasets used by the BigDataOcean project, some of the instances are from the Climate and Forecast (CF) Standard Names.<sup>2</sup>

<sup>1</sup><http://marineregions.org/>

<sup>2</sup><http://cfconventions.org/Data/cf-standard-names/60/build/cf-standard-name-table.html>

```

{
  uri: "http://www.bigdataocean.eu/standards/canonicalmodel/ontology.xml",
  nsp: "http://www.bigdataocean.eu/standards/canonicalmodel#",
  prefix: "bdo",
  - titles: [
    - {
      _id: "5b88558ea1455b0c102b55ae",
      value: "Canonical Model: Standard Variables for Big Data Ocean",
      lang: "en"
    }
  ],
  + descriptions: [...],
  + tags: [...],
  + pilots: [...],
  lastModifiedInLOVAt: "2018-11-27T16:35:28.324Z",
  issuedAt: "2018-08-15T00:00:00.000Z",
  lastDeref: "2018-08-30T20:34:47.057Z",
  + creatorIds: [...],
  contributorIds: [ ],
  + publisherIds: [...],
  + reviews: [...],
  - versions: [
    - {
      isReviewed: true,
      + languageIds: [...],
      + relMetadata: [...],
      relDisjunc: [ ],
      + relEquivalent: [...],
      + relExtends: [...],
      relGeneralizes: [ ],
      relImports: [ ],
      relSpecializes: [ ],
      name: "v2018-08-30",
      fileURL: "http://212.101.173.48:3333/dataset/bdo/vocabs/bdo/versions/2018-08-30.n3",
      issued: "2018-08-30T20:34:47.057Z",
      classNumber: "15",
      propertyNumber: "4",
      instanceNumber: "344",
      datatypeNumber: "0",
      reviewed: false
    }
  ]
}

```

FIGURE 4.3: JSON file of the result from executing the Info Vocab API of Canonical Model Vocabulary.

## 4.2 BigDataOcean Harmonization Tool

The **BigDataOcean Harmonization Tool** goal is to semantically interpret datasets employing appropriate ontologies and vocabularies related to the BDO Vocabulary Repository. The Harmonization Tool describes dataset metadata using terms and instances from the ontologies/vocabularies saved in RDF triples.

The main functionalities of the BDO Harmonization Tool are the following:

- Enabling the user to import metadata from the different datasets pilots of the BDO
- Allowing the user to view, insert, update, and delete metadata for a specific dataset
- Facilitating the user to navigate to linked vocabularies using SPARQL or RESTful APIs.

BDO Harmonization Tool can extract metadata from datasets written in different formats, such as NetCDF, CSV, and Excel; as well as datasets available by external services,

like the Copernicus service<sup>3</sup>. Eventually, all metadata of the BDO datasets stored in the Apache Jena Fuseki Triple Store<sup>4</sup>, and it can be queried and retrieved using one of two functionalities offer by the BDO Harmonization Tool, the APIs, or the SPARQL endpoint.

While extracting metadata from a new dataset file, it is possible to obtain unseen data. These data refer to the information not provided inside the dataset. In the case of unseen data, the BDO Harmonization Tool requires the input of domain experts to perform the annotation of the data attributes that were not imported into the BDO Harmonization Tool. The domain expert has the role of choosing from the BDO Vocabulary Repository which instances are the best match according to the meaning of the attributes in the dataset.

Figure 4.4 highlights the manual interaction flow of the BDO Harmonization Tool. This flow can break down into two points:

1. **Select vocabularies:** A domain expert selects manually the vocabularies terms and instances, e.g., `dct:title`<sup>5</sup>, `dcats:subject`<sup>6</sup>, which describes the metadata of the dataset. Those vocabularies are saved manually in JSON<sup>7</sup> files inside the BDO Harmonization Tool.
2. **Select dataset:** A domain expert (user) selects the dataset to add. BDO Harmonization Tool requests the file to the BDO HDFS<sup>8</sup> via API, and then the BDO HDFS returns the raw data. Then BDO Harmonization Tool extracts metadata from the raw file. It is important to note that some metadata attributes could be empty (unseen data), in this case, the domain expert will annotate the missing data and confirm the mappings for the suggested variables fields, finally saves the dataset metadata into the BDO Harmonization triple store.

The metadata contains essential information like identifier, title, description, subject, keyword, geographic location and variables used in the dataset. Figure 4.5 presents the dataset metadata structure model stored as RDF data in the BDO Harmonization Triple Store. The BDO Harmonization dataset metadata model has a similar structure to the Data Catalog Vocabulary (DCAT). DCAT<sup>9</sup> defined as a “collection of data, published or curated by a single agent, and available for access or download in one or more formats.”

<sup>3</sup><http://marine.copernicus.eu/services-portfolio/access-to-products/>

<sup>4</sup><https://jena.apache.org/documentation/fuseki2/>

<sup>5</sup>DCMI Metadata Terms Vocabulary (dct)

<sup>6</sup>Data Catalog Vocabulary (dcat)

<sup>7</sup>JavaScript Object Notation (JSON)

<sup>8</sup>HDFS stands to Hadoop Distributed File System

<sup>9</sup><https://www.w3.org/TR/vocab-dcat/>

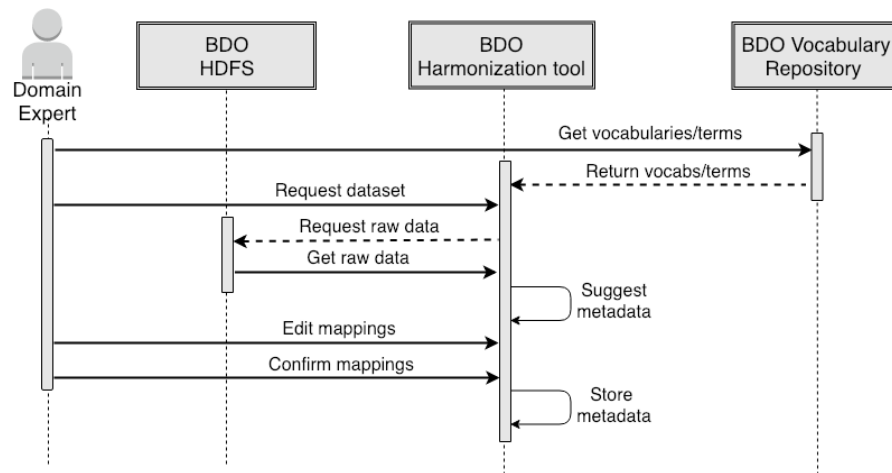


FIGURE 4.4: Flowchart Diagram BigDataOcean Harmonization Tool.

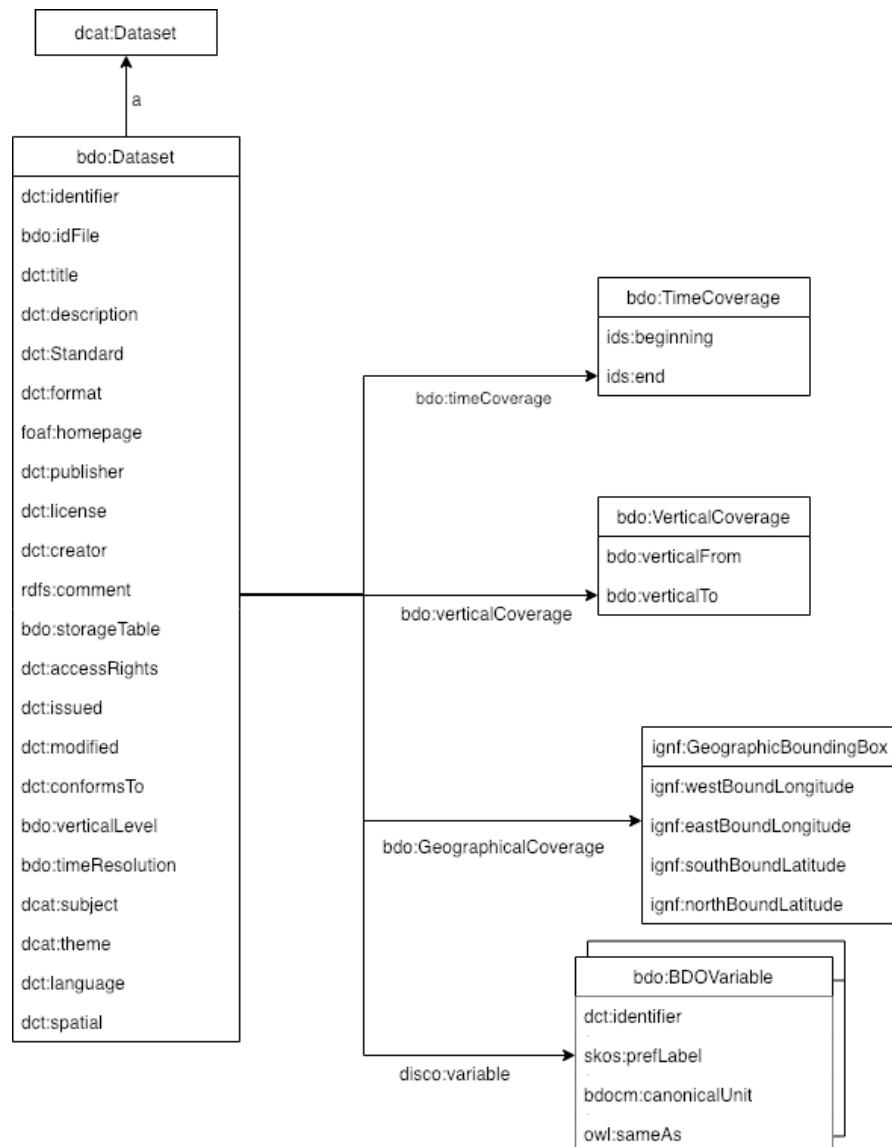


FIGURE 4.5: BigDataOcean Harmonization Dataset Metadata Structure Model.

Listing 4.1 shows the dataset metadata of Med Sea - NRT in situ Observations saved in BigDataOcean Harmonization Tool.

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX disco: <http://rdf-vocabulary.ddialliance.org/discovery#>
PREFIX dcat: <https://www.w3.org/TR/vocab-dcat/>
PREFIX bdo: <http://bigdataocean.eu/bdo/>
PREFIX bdocm: <http://www.bigdataocean.eu/standards/canonicalmodel#>
PREFIX ids: <http://industrialdataspace/information-model/>
PREFIX qudt: <http://qudt.org/schema/qudt/>
PREFIX ignf: <http://data.ign.fr/def/ignf#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX inspire: <http://inspire.ec.europa.eu/metadata-codelist/
  TopicCategory/>
PREFIX eionet: <https://www.eionet.europa.eu/gemet/en/concept/>

bdo:MO_201801_TS_MO_ATHOS a dcat:Dataset ;
dct:identifier "MO_201801_TS_MO_ATHOS" ;
bdo:idFile "" ;
dct:title "Med Sea – NRT in situ Observations" ;
dct:description "HCMR Copernicus Insitu" ;
dct:Standard "CF-1.6 OceanSITES-Manual-1.2 Copernicus-InSituTAC-SRD
  -1.3 Copernicus-InSituTAC-ParametersList-3.1.0" ;
dct:format "NetCDF" ;
foaf:homepage "http://www.hcmr.gr" ;
dct:publisher "OceanSITES" ;
dct:license "See Copernicus Marine Environment Monitoring Service
  Data commitments and licence at: http://marine.copernicus.eu/web
  /27-service-commitments-and-licence.php" ;
dct:creator "moored surface buoy" ;
rdfs:comment "Time-series" ;
bdo:storageTable "hcmr_timeseries" ;
dct:accessRights "Public" ;
dct:issued "2018-07-02T08:10:14"^^xsd:dateTime ;
dct:modified "2018-09-28T09:16:32"^^xsd:dateTime ;
bdo:GeographicalCoverage bdo:MO_201801_TS_MO_ATHOS_GC ;
dct:conformsTo "" ;
bdo:verticalCoverage bdo:MO_201801_TS_MO_ATHOS_VC ;
bdo:verticalLevel "" ;
bdo:timeCoverage bdo:MO_201801_TS_MO_ATHOS_TC ;

```

```

bdo:timeResolution "daily" ;
dcat:subject inspire:climatologyMeteorologyAtmosphere;
dcat:theme eionet:7402 ;
dct:language "eng";
disco:variable bdo:MO_201801_TS_MO_ATHOS_LATITUDE ,
               bdo:MO_201801_TS_MO_ATHOS_DRYT_DM .

bdo:MO_201801_TS_MO_ATHOS_LATITUDE a bdo:BDOVariable ;
    dct:identifier "LATITUDE" ;
    skos:prefLabel "latitude"@en ;
    bdocm:canonicalUnit "degree_north" ;
    owl:sameAs bdocm:latitude .

```

LISTING 4.1: A snippet of the Med Sea - NRT in situ Observations dataset metadata in Turtle format

Table 4.1 presents the Harmonization dataset metadata attributes which are linked with vocabularies from the BDO Vocabulary Repository. Here, there are four attributes from the BDO Harmonization Tool which have a link with specific vocabulary.

Attribute from Harmonization Tool	Definition	Linked vocabulary
Subject	The general interpretation of the dataset	INSPIRE Schema Vocabulary <sup>10</sup>
Keywords	The specific interpretation of the dataset	GEMET definitions schema Vocabulary <sup>11</sup>
Geographic Location	Place interpretation of the dataset	Geographic Location Standards for BigDataOcean <sup>12</sup>
Variable	Definition of the standard variable	Canonical Model: Standard Variables for BigDataOcean <sup>13</sup>

TABLE 4.1: Harmonization dataset metadata linked with vocabularies.

<sup>10</sup><http://inspire.ec.europa.eu/metadata-codelist/TopicCategory/TopicCategory.en.rdf>

<sup>11</sup><https://www.eionet.europa.eu/gemet/exports/latest/gemet-definitions.rdf>

<sup>12</sup><http://www.bigdataocean.eu/standards/geographiclocation/index.html>

<sup>13</sup><http://www.bigdataocean.eu/standards/canonicalmodel/index.html>

## Chapter 5

# Methodology

This chapter presents the integration details of the automated linked discovery tools on the BDO project which will help to extend and automatize the process when extracting the metadata of a dataset.

Based on the comparison of NLP, we chose Stanford Named Entity Recognizer tool, due to, it can extract all tags specified by a single model and has better accuracy and time performance than Apache OpenNLP. For the interlinking data, we chose LIMES tool for the reason that it offers a RESTful API and it is faster by reducing the number of comparisons needed during mapping.

Figure 5.1 presents two components, the left component is the BigDataOcean project which contains the BDO Vocabulary Repository and the BDO Harmonization Tool, and the right component is the Automated Link Discovery containing Stanford NER, and LIMES tools are.

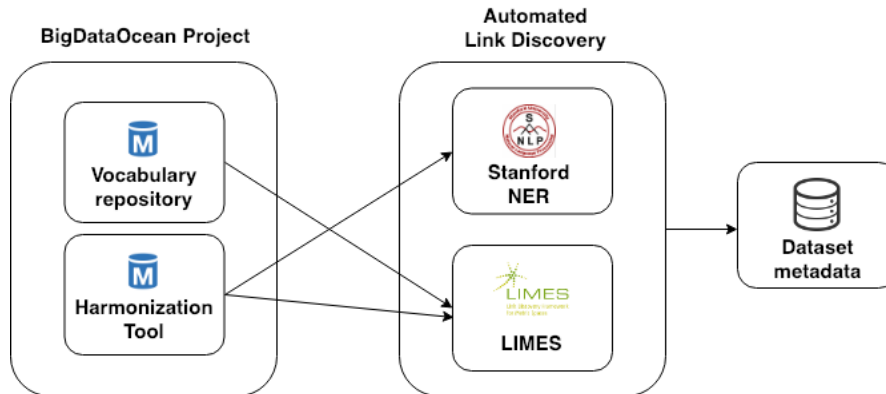


FIGURE 5.1: Methodology workflow diagram

The integration process is simple, BDO project connects with Automated Link Discovery to output complete dataset metadata without the interaction of the user. To use LIMES,



it needs to have access to BDO Vocabulary Repository and BDO Harmonization Tool to use the vocabularies and the metadata which exists in the dataset while Stanford NER only connects to the BDO Harmonization Tool to extract some metadata such as title, description and location.

## 5.1 Architecture Implementation

BDO Harmonization Tool connects with BDO Vocabulary Repository to extract the vocabularies explained in table 4.1 to be used later in the Automated Link Discovery tools.

To extend the BDO Harmonization Tool, we created two main pipelines. The reason behind is that we cannot extract named entities for raw variables while in the case of the title, description, and location we can obtain some named entities which are going to link with a vocabulary using LIMES, see Figure 5.2.

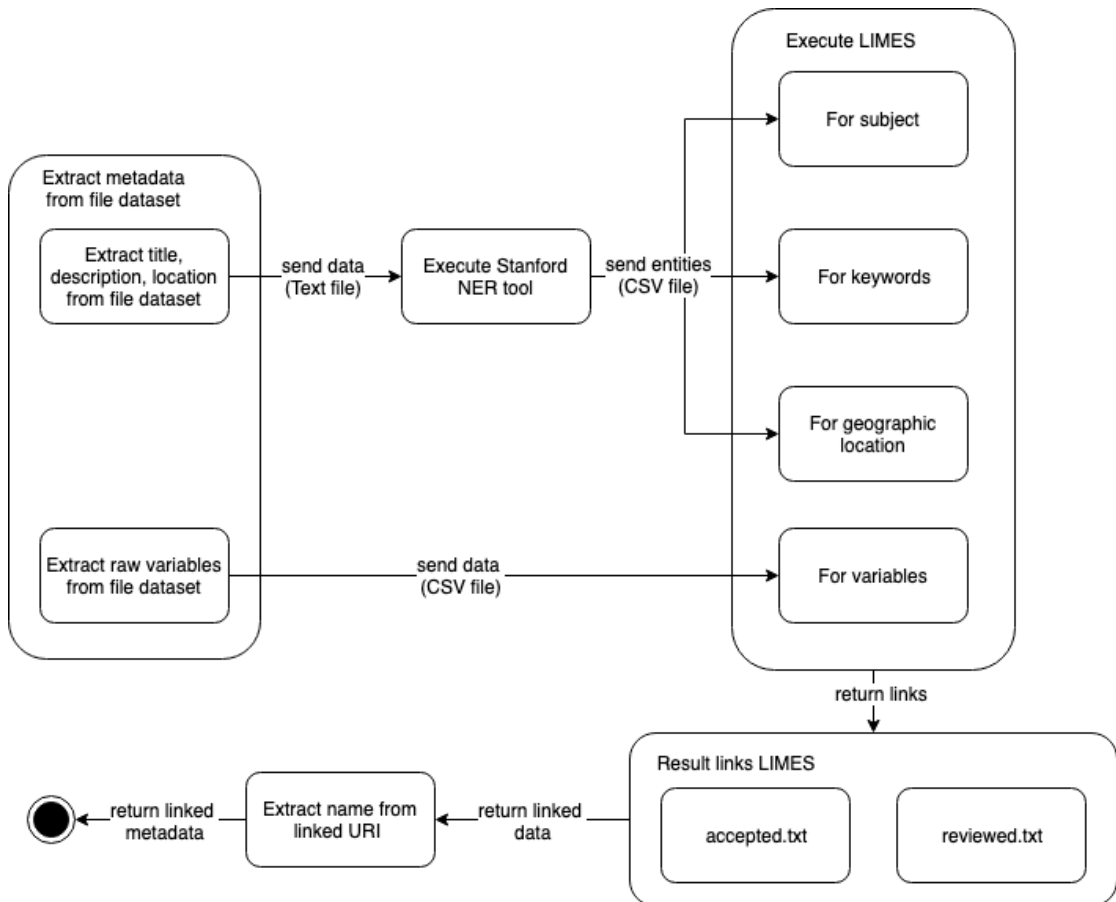


FIGURE 5.2: Pipeline Diagram for the creation of the Automated Link Discovery process.

Before going into detailed about the description of each pipeline, we created an example using a dataset from Copernicus see Figure 5.3. In the figure, we present a general idea of the pipeline diagram. As explained before there are two main pipelines.

The first pipeline will extract from the dataset, the title, description, and location, in order to obtain the entities using the Stanford NER tool; later those entities are going to be linked with LIMES tool using the vocabularies INSPIRE, GEMET and Geographic Location to identify linked entities for the subject, keywords, and geographic location attributes. e.g., Subject: Oceans.

The second pipeline will extract only the list of variables from the dataset which are going to be linked with the Canonical Model vocabulary using the LIMES tool. e.g., the variable `sea_water_salinity` represents salinity.

When each pipeline the process finishes, it will return the corresponding information to each attribute.

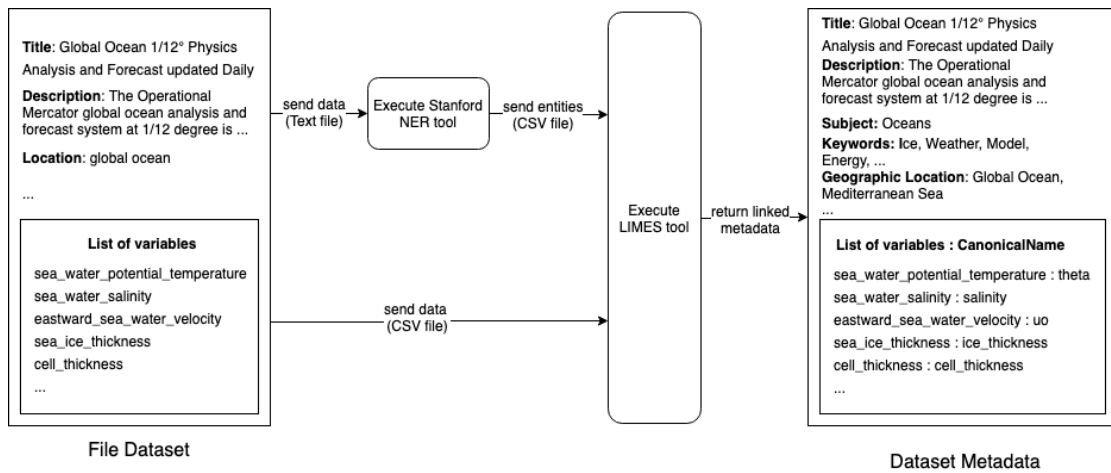


FIGURE 5.3: Example of Pipeline Diagram for the creation of the Automated Link Discovery process.

The first pipeline is to save the title, description, and location in a text file where Stanford NER will read and classify the words into entities. When the process finished and the BDO Harmonization Tool made the post-processing method, as explained in section 5.2.1.1, the entities are saved in a CSV file where the header name is “foaf:name”<sup>1</sup>. Next, Harmonization Tool will create three new pipelines, one for the subject, one for keywords and the last one for geographic location. Each of the pipelines creates a LIMES configuration file in XML, as explained in section 5.2.2. Finally when LIMES returns the links in *accepted.txt* and *reviewed.txt* files, then the Harmonization Tool extracts the links and converts the URIs given into the URL (e.g., geographic location

<sup>1</sup>foaf:name is same as “http://xmlns.com/foaf/0.1/name”

URL saved in the term `skos:closeMatch`<sup>2</sup>) using Apache Jena, see `OntologyAnalyser` and `SPARQLRunner` methods in Figure 5.4.

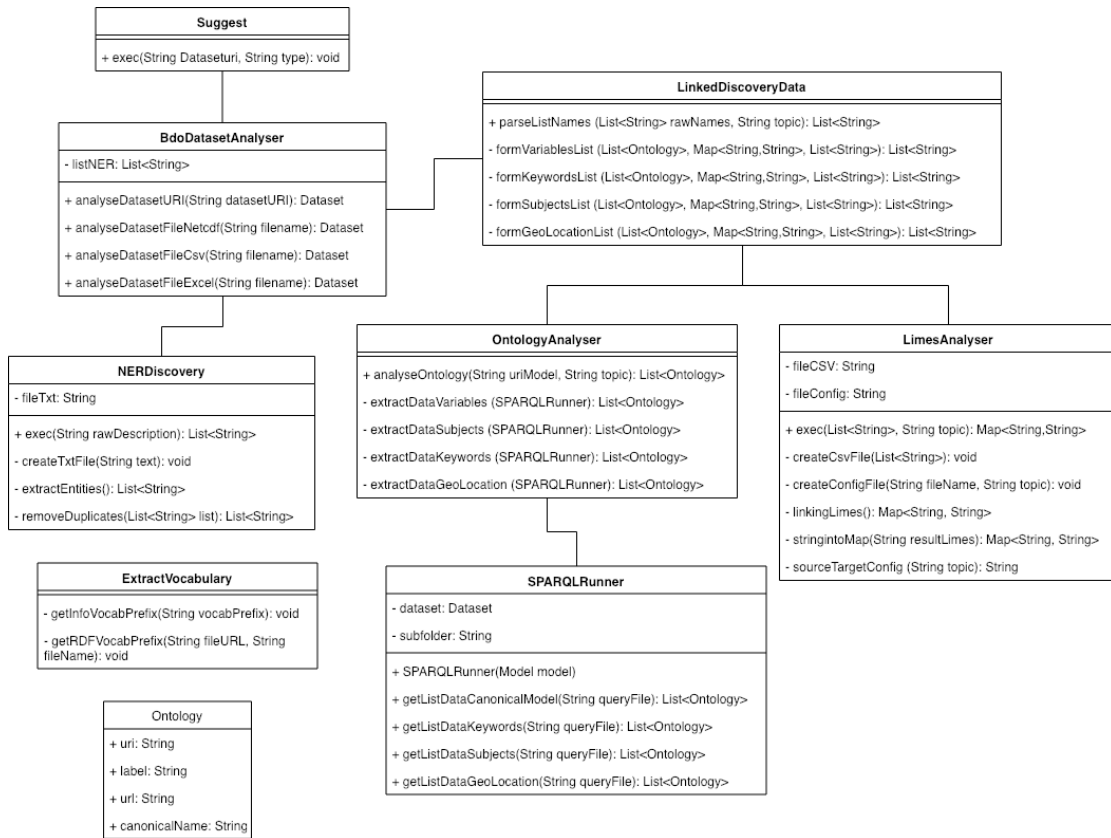


FIGURE 5.4: BigDataOcean Harmonization Tool class diagram.

The second pipeline is for raw variables, where the BDO Harmonization Tool will create a CSV file with the raw variables given in the dataset file and adding the header name “`http://xmlns.com/foaf/0.1/name`”. Next, the BDO Harmonization Tool creates the LIMES configuration and execute LIMES. When the process is finished, the Harmonization Tool will do the same step as in the first pipeline, but instead of extracting the URL it will extract the canonical Name from the term `bdo:hasCanonicalName`<sup>3</sup>. At the same time, if LIMES identifies that the raw variable has more than one link then the BDO Harmonization Tool, will create a list of suggestions.

BDO Canonical Model vocabulary (bdo) is used for variables, there can find the same name repeated many times. The reason is that a name is not unique. Therefore, it can represent more than one canonical name. A canonical name is a standard name taken from the Climate and Forecast (CF) Metadata Conventions<sup>4</sup> or created by the BDO project. E.g., the name *direction* has two canonical names, *profile\_direction* or

<sup>2</sup><https://www.w3.org/2008/05/skos>

<sup>3</sup><http://www.bigdataocean.eu/site/canonicalmodel/>

<sup>4</sup><http://cfconventions.org/>

*platform\_true\_heading*. Consequently, all the canonical names which LIMES linked are possible matching for a specific raw variable.

When the two processes finished, Harmonization Tool will output the dataset metadata.

The architecture diagram of the BDO Harmonization Tool is divided into two layers, the frontend, and the backend, see Figure 5.5. In this section, we only focused on the backend layer where the Automated Link Discovery process takes place. As explained before, the pipelines will extract the information like title, description, location, and raw variables from the dataset metadata that later will be passed to Stanford NER and LIMES tools to identify and map the linked entities for the subject, keywords, geographic location, and variables attributes in the BDO Harmonization Tool.

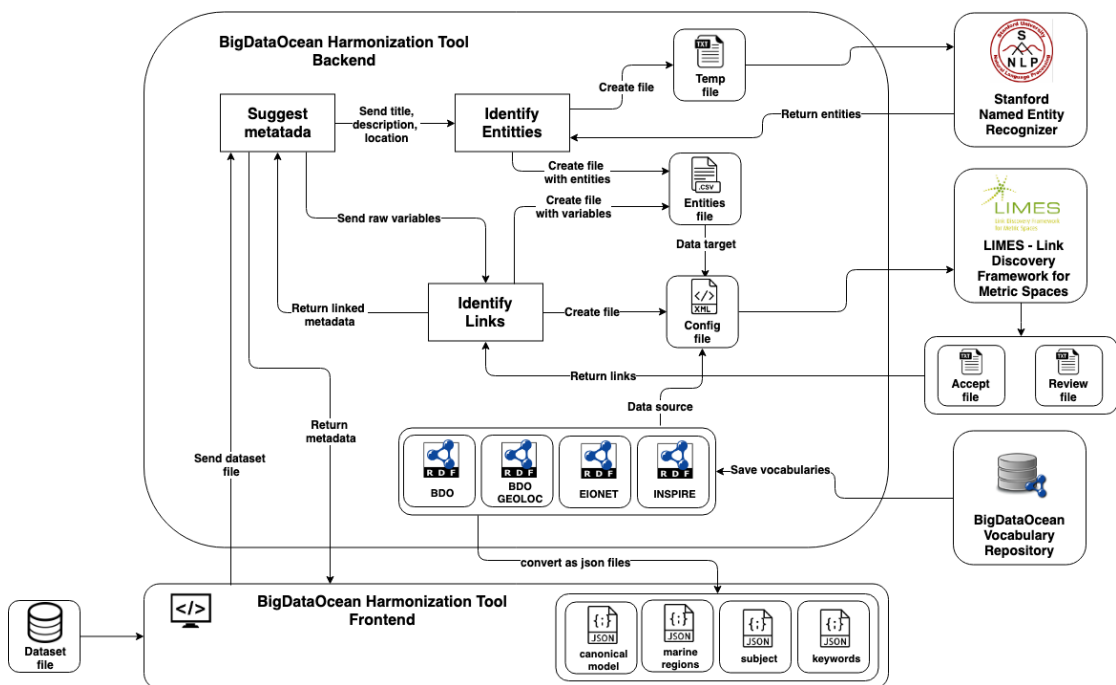


FIGURE 5.5: Architecture Diagram.

## 5.2 Automated Link Discovery Implementation

As discussed in section 4.2, BDO Harmonization Tool cannot extract all the metadata from the datasets (unseen data). Therefore, with the help of Stanford NER and LIMES, BDO Harmonization Tool can automatically extract entities and interlink instances to obtain missing metadata for four mandatory fields/attributes.

In this section, a detailed explanation of the implementation process using Stanford NER and LINES into BDO Harmonization Tool will happen. BDO Harmonization Tool

works with the Stanford Named Entity Recognizer 3.9.2 version and with the LINES 1.5.0 version (currently latest published version).

### 5.2.1 Stanford Named Entity Recognizer

Stanford Named Entity Recognizer (Stanford NER) needs a text to identify and classify each of the words into entities. BDO Harmonization Tool creates a temporary text file which includes the title, the description, and the location of the dataset. Then Stanford NER Tool with the help of the two different classifier models suggests the corresponding class in each word, see table 5.1. The classifier model *english.all.3class* taken from the Stanford NER tool in which the model identifies the location, person, and organization entities, see section 3.2.1.1. Also, the *bdo\_model\_ner* is a classifier model explicitly created for this master thesis, see section 5.2.1.2.

Classifier	Classes
english.all.3class	Location, person, organization
bdo_model_ner	Subject, keywords, misc

TABLE 5.1: BigDataOcean Harmonization Tool models classifiers

#### 5.2.1.1 Post-processing of Stanford NER

When BDO Harmonization Tool executes the Stanford NER tool, it returns the entities which were identified by the two models as a list. If two entities one followed by the other, then Stanford NER returned it as a single entity, e.g., Mediterranean Sea. However, sometimes the text does not contain any connectors or prepositions between the entities giving as a result of a single entity. Table 5.2 presents example two texts, and the result given by Stanford NER.

Text	Entities
weekly mean fields from Global Ocean Biogeochemistry Analysis	Global Ocean Biogeochemistry Analysis
Global Ocean - In Situ Observation Copernicus	Global Ocean, In Situ

TABLE 5.2: BigDataOcean Harmonization Tool models classifiers

BDO Harmonization Tool does a post-processing method to give better entities results. Figure 5.6 presents the process when Stanford NER return the entity. In this case, the process is divided into four steps. The first step saves the entity given by Stanford

NER; then the second step verifies if the entity in the first step contains spaces, then it separates the entity into multiple entities. The third step checks if the entity in the first step has more than two spaces, then it creates a combination of entities. The last step is to group all the results and then create a list without entity duplicates.

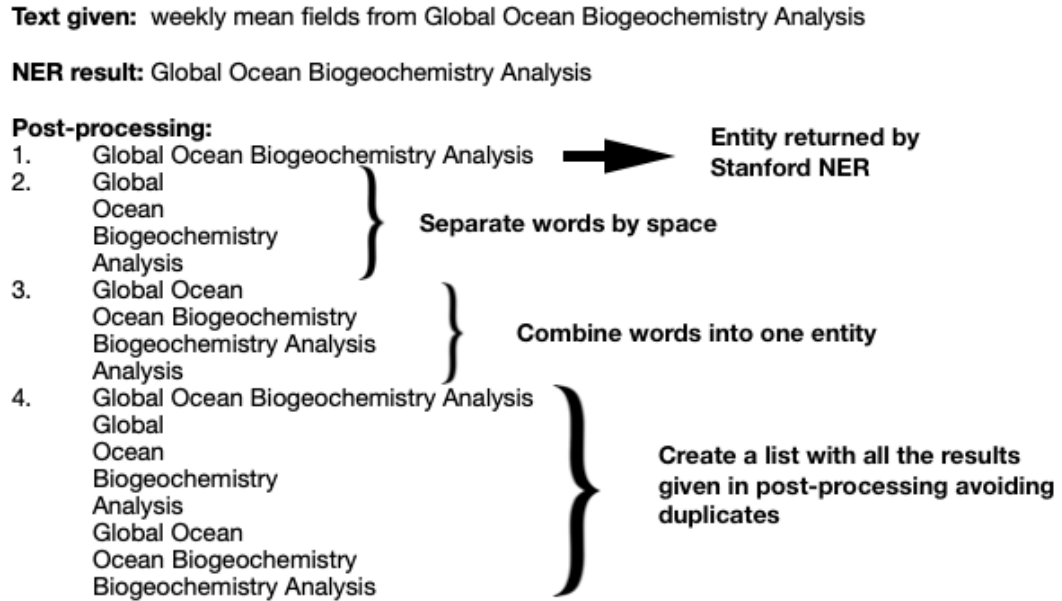


FIGURE 5.6: Training Data of Named Entity Recognition.

### 5.2.1.2 Training Data

As explained in the last section, BDO Harmonization Tool uses two classifier models. The classifier models which Stanford offers are not sufficient for this master thesis, because the models cannot identify specific named entities as oceans, forecasting, temperature, and so on. Therefore, it was necessary to create the model *bdo\_model\_ner* to help identify correctly the entities which then BDO Harmonization Tool will use in the next process.

Figure 5.7 presents the training model creation where the process must be created manually. The training model for the Stanford NER tool is a supervised training model. A supervised training model analyzes the training data and produces an inferred function, which is used for mapping new examples [39].

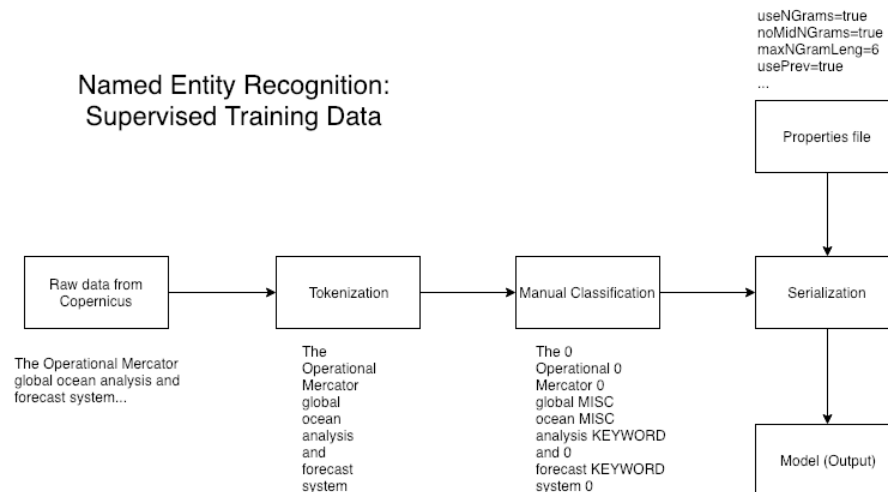


FIGURE 5.7: Training Data of Named Entity Recognition Flow Chart.

The primary step is to find a good quantity of training data, in this case, BDO project has lots of open datasets which are used for the training. With the help of Stanford NER tool, it tokenized the text. Then manually, each word classified as SUBJECT, KEYWORD or MISC, separated by a TAB, if the word has no classification then it needs to add O. When the process finished and all the words have a classification, Stanford NER serialized the training data into a model given a configuration. Listing 5.1 shows a snippet of the configuration.

```

# location of the training file
trainFile = bdo_model_ner.tsv
# location where you would like to save (serialize) your
# classifier; adding .gz at the end automatically gzips the file ,
# making it smaller , and faster to load
serializeTo = bdo_model_ner.ser.gz

# structure of your training file; this tells the classifier that
# the word is in column 0 and the correct answer is in column 1
map = word=0,answer=1
# This specifies the order of the CRF: order 1 means that features
# apply at most to a class pair of previous class and current class
# or current class and next class .
maxLeft=1
# these are the features we'd like to train with
# some are discussed below , the rest can be
# understood by looking at NERFeatureFactory
useClassFeature=true
useWord=true
# word character ngrams will be included up to length 6 as prefixes
# and suffixes only

```

```
useNGrams=true
noMidNGrams=true
maxNGramLeng=6
```

LISTING 5.1: Snippet of training configuration

## 5.2.2 LIMES

As explained in section 3.2.2.1, LIMES needs a configuration file to execute the tool. The configuration file divided into five parts, such as metadata, prefixes, source/target data sources, metrics for similarity measurement, acceptance/review conditions, and output format. In this section, only three parts are going to vary for each of the four attributes in BDO Harmonization Tool.

### 5.2.2.1 Data Sources

LIMES estimates links between two data sources denominated target and source. Both data sources have a configuration which helps interlink correctly. For this master thesis, LIMES configuration will be as follows:

For target,

- The **ENDPOINT** will be the CSV file created with the entities extracted by Stanford NER or with the raw variables given by the dataset file.
- The **RESTRICTION** will remain focused in the column of the CSV file which header name is *foaf:name*.
- For the **PROPERTY** will transform the data into lowercase.
- Last, the **TYPE** will be CSV.

For source,

- The **ENDPOINT** will depend on the pipeline that the BDO Harmonization tool is running. For variables, the endpoint is Canonical Model vocabulary (*bdo*). For geographic location, the endpoint is Geographic Location vocabulary (*bdogeoloc*). For keywords, the endpoint is GEMET definition vocabulary (*eionet*). For subjects, the endpoint is INSPIRE schema (*inspire*).
- Each vocabulary has its restriction and properties, therefore the **RESTRICTION** and **PROPERTY** will depend on the pipeline.



- Last, the **TYPE** will be N3.

Listing 5.2 presents an example of the target configuration. In the case of source configuration, the listing 5.3 shows the variables pipeline where there are two properties in which LIMS will remove the language tags and transform the data in lowercase.

```
<TARGET>
<ID>RawNames</ID>
<ENDPOINT>/BDOHarmonization/BigDataOcean-Harmonization/Backend/
  AddDatasets/temp.csv</ENDPOINT>
<VAR>?x</VAR>
<PAGESIZE>1000</PAGESIZE>
<RESTRICTION>?x rdf:type http://xmlns.com/foaf/0.1/name</RESTRICTION>
<PROPERTY>http://xmlns.com/foaf/0.1/name AS lowercase</PROPERTY>
<TYPE>CSV</TYPE>
</TARGET>
```

LISTING 5.2: LIMS target data configuration example

```
<SOURCE>
<ID>OntologyRDF</ID>
<ENDPOINT>/dataHarmonization/ontologiesN3/bdo.n3</ENDPOINT>
<VAR>?y</VAR>
<PAGESIZE>1000</PAGESIZE>
<RESTRICTION>?y a bdo:Variable</RESTRICTION>
<PROPERTY>rdfs:label AS nolang->lowercase</PROPERTY>
<PROPERTY>bdo:hasCanonicalName AS nolang->lowercase</PROPERTY>
<TYPE>N3</TYPE>
</SOURCE>
```

LISTING 5.3: LIMS source data configuration for the pipeline variables

A different source configuration example is in the subject pipeline where there is no restriction, but LIMS will remove the language tags and transform the data in lowercase, see listing 5.4.

### 5.2.2.2 Selection of Metrics

In section 3.2.2.1.4, LIMS defined the metric as one of the essential steps of using a link discovery implementation. The metric defines the measures distance between two strings for approximate string matching or comparison [40]. In other words, the metrics are the measure of how much equals two data sources are. As explained in the section

```

<SOURCE>
<ID> OntologyRDF </ID>
<ENDPOINT> /dataHarmonization/ontologiesN3/inspire.n3 </ENDPOINT>
<VAR> ?y </VAR>
<PAGESIZE> 1000 </PAGESIZE>
<RESTRICTION></RESTRICTION>
<PROPERTY> dct:title AS nolang->lowercase </PROPERTY>
<TYPE> N3 </TYPE>
</SOURCE>

```

LISTING 5.4: LINES source data configuration for the pipeline subjects

LINES will only operate with string measures package. Therefore, a previous analysis was done to see how reliable are the metrics for link matching, see section 6.2.

After experimenting with the different metrics, cosine similarity and Qgrams were the most successful metric for identifying and linking equivalent name strings. Listing 5.5 shows an example of a metric definition in LINES for the pipeline variables. In this example, the metric works with the two categories which LINES provides, metrics operation as *Cosine* and boolean operation as *OR*. When the metric tag is working with boolean operations, BDO Harmonization Tool needs to specify the two children, in this case, it compares between the name or the canonical name of the variable; also, it is essential to add the threshold in both children.

```

<METRIC>
  OR( Cosine(x.foaf:name, y.rdfs:label) | 0.8 , Cosine(x.foaf:name, y.
    bdo:hasCanonicalName) | 0.8 )
</METRIC>

```

LISTING 5.5: Metric configuration definition in LINES for variables

### 5.2.2.3 Acceptance and Review Conditions

Acceptance and review conditions consist of setting the threshold value which is the minimum value that the two instances must have to identify the relation correctly. LINES will operate with various threshold values depending on the work pipeline.

For the case of variables pipeline, the acceptance and review threshold value are near to 1. If the raw variable and the canonical name have the same string name, then they ought to be the same variable. LINES accept the link as correct, which will output the metric value higher than 0.98. Listing 5.6 shows the acceptance condition in LINES for the variables pipeline.

```

<ACCEPTANCE>
  <THRESHOLD> 0.98 </THRESHOLD>

```

```

<FILE> accepted.txt </FILE>
<RELATION> owl:sameAs </RELATION>
</ACCEPTANCE>

```

LISTING 5.6: Acceptance condition in LIMES for variables pipeline

Listing 5.7 shows the review condition for variables pipeline. Here, LIMES will review a link when the threshold value is greater than 0.95 and less than the acceptance condition (0.98).

```

<REVIEW>
  <THRESHOLD> 0.95 </THRESHOLD>
  <FILE> reviewme.txt </FILE>
  <RELATION> owl:sameAs </RELATION>
</REVIEW>

```

LISTING 5.7: Review condition in LIMES for variables pipeline

In this master thesis, the BDO Harmonization Tool will verify if there are acceptance links. When the acceptance is empty, Harmonization Tool will take the links from the review condition.

### 5.3 Front-end Implementation

As explained in section 5.1, BDO Harmonization Tool connects with BDO Vocabulary Repository via API to extract the vocabularies. This process is done every year to download the latest version of each vocabulary. Once the vocabularies (N3 format) are saved in the system, BDO Harmonization Tool converts the vocabularies into JSON format to later use them in the frontend.

Figure 5.8 shows the sequence diagram of the BDO Vocabulary Repository and BDO Harmonization Tool integrated with the Automated Link Discovery. Here, the user connects to BDO Harmonization Tool frontend in which presents an HTML page where the user can select the dataset file from a list to upload into the system, see Figure 5.9. Then BDO Harmonization will call the BDO HDFS to retrieve a copy of the dataset file. Next, the BDO Harmonization Tool frontend will call the backend to extract the metadata of the dataset file. While extracting the metadata from the dataset file, the BDO Harmonization Tool will connect with Stanford NER and LIMES tools to automatically suggest missing metadata.

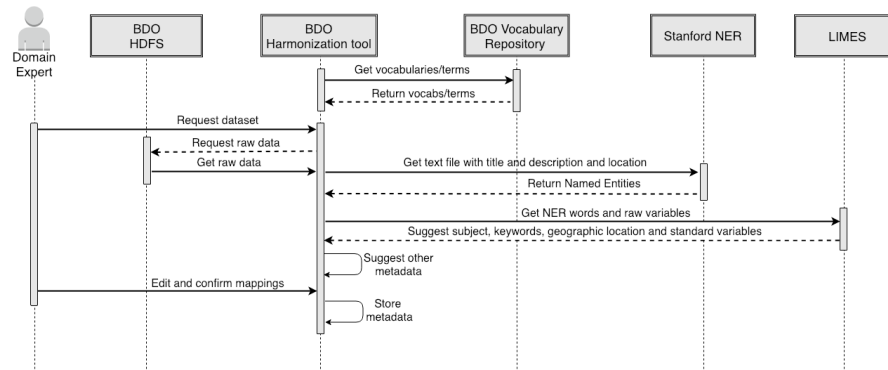


FIGURE 5.8: Flowchart diagram of BigDataOcean Harmonization Tool using Vocabulary Repository, Stanford NER and LINES

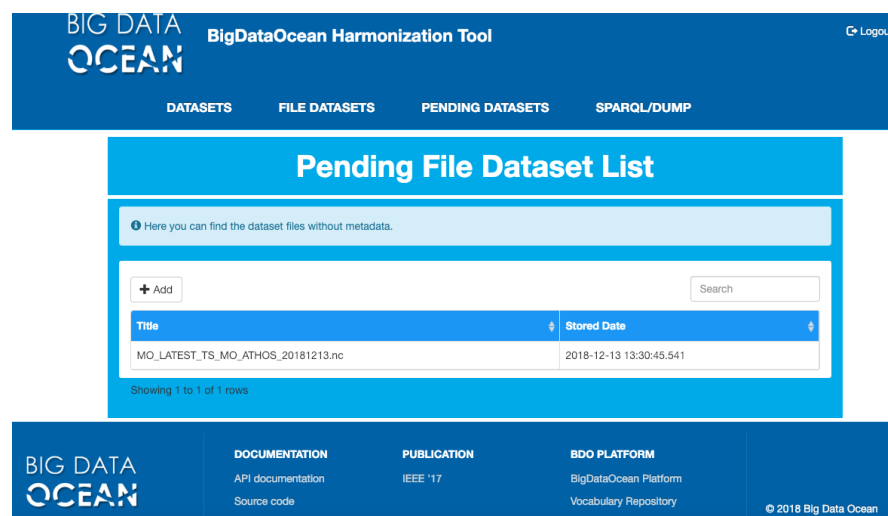


FIGURE 5.9: Pending dataset files to be added into Harmonization Tool.

As explained in figure 5.2, in order to suggest missing metadata the BDO Harmonization Tool needs to connect to two main pipelines, one for suggesting the subject, keywords, and geographic locations metadata and the other for suggesting the canonical name for the variables on the dataset file. It is important to note that LINES can identify that the raw variable has more than one link then the Harmonization Tool will create a list of suggestions that later in the frontend, the user will manually select the correct canonical name.

When the two pipelines processes finish, Harmonization Tool will return the dataset metadata to the frontend. Figure 5.10 shows that for this dataset file there is one subject and three keywords, one geographic location and for the variables, there are two variables which have more than one canonical name (DIRECTION, and LATITUDE). As explained before, the user manually will select one of the options and automatically the field will be filled in.

SaveCancel

\* These fields are required!

Title:

Arctic- NRT in situ Observations

Description:

Subject:

http://inspire.ec.europa.eu/metadata-codelist/TopicCategory/oceans

Keywords:

https://www.eionet.europa.eu/gemet/en/concept/5769

https://www.eionet.europa.eu/gemet/en/concept/5694

https://www.eionet.europa.eu/gemet/en/concept/4359

Standards:

CF-1.6 OceanSITES-Manual-1.2 Copernicus-InSituTAC-SRD-1.3 Copernicus-InSituTAC-ParametersSL

Data Format:

Geographic Location:

http://marineregions.org/mrgid/1912

Geographical Coverage:

West:

1.8935

East:

20.0647

South:

56.5827

North:

73.9925

+ Add Row

Dataset Variable	BDO Variable	Unit Variable	Action
DIRECTION	<div>direction: profile_direction</div> <div>direction: platform_true_heading</div>		
CNDC	sea_water_electrical_conductivity	S m-1	
LATITUDE	<div>latitude: latitude</div> <div>lat: latitude</div>	degree_north	
TEMP	sea_water_temperature	degrees_C	
POSITION_QC	position_quality_flag		
PSAL_QC	sea_water_practical_salinity_quality_flag		
TEMP_QC	sea_water_temperature_quality_flag		

FIGURE 5.10: Snippet of the frontend where metadata suggested using Stanford NER and LIMES

Finally, figure 5.11 presents the dataset metadata saved in the BDO Harmonization Tool.

Arctic- NRT in situ Observations

(AR\_201206\_PR\_CT\_MYO\_AR\_58GS)

Description:

this is a description

Subject:

Oceans

Keywords:

North Atlantic Oceanoceanin situ

Standards:

CF-1.6 OceanSITES-Manual-1.2 Copernicus-InSituTAC-SRD-1.3 Copernicus-InSituTAC-ParametersList-3.1.6

Data Format:

NetCDF

Language:

en

Homepage:

Homepage Link

Geographic Location:

North Atlantic Ocean

Dataset Variable	BDO Variable	Unit Variable
DIRECTION	profile_direction	-
CNDC	sea_water_electrical_conductivity	S m-1
LATITUDE	latitude	degree_north
TEMP	sea_water_temperature	degrees_C
POSITION_QC	position_quality_flag	-
PSAL_QC	sea_water_practical_salinity_quality_flag	-
TEMP_QC	sea_water_temperature_quality_flag	-

Suggested by Stanford NER and LIMES

Suggested by LIMES

FIGURE 5.11: Snippet of dataset metadata saved in BigDataOcean Harmonization Tool

## Chapter 6

# Evaluation

This chapter presents the BigDataOcean Harmonization Tool evaluation which is divided into four parts. The first section presents the datasets used for the accuracy and performance evaluation testing it with a gold standard. The second section relates to the assessment of the string similarity measures used by LINES. The third section presents the accuracy evaluation of using Stanford Named Entity Recognizer and LINES in the BigDataOcean Harmonization Tool. The last section shows the BigDataOcean Harmonization Tool performance evaluation.

### 6.1 Datasets

We build a gold standard dataset by collecting open source datasets with the different permitted type of formats, such as Copernicus, NetCDF, CSV, and Excel, by BigDataOcean Harmonization Tool.

The gold standard dataset is linked with the four vocabularies explain in table 4.1. Figure 6.1 shows that for each file dataset we extract the title, description, location (area) and the raw variables, later, we manually classify and identify the subject, keywords, geographic location, and variables using the vocabularies INSPIRE, GEMET, Geographic Location, and Canonical Model respectively.

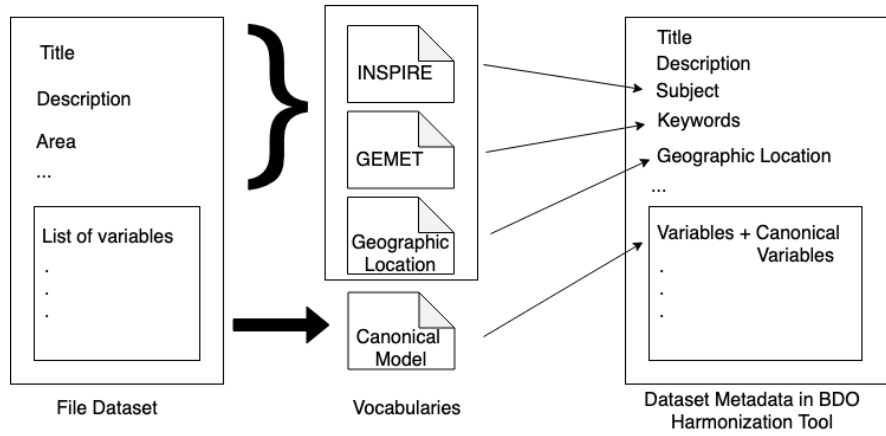


FIGURE 6.1: Representation of the gold standard

For the gold standard dataset, we collect a total of eleven heterogeneous datasets in which six are from Copernicus, three are NetCDF, one CSV, and one Excel. Table 6.1 presents the title, the text length, the number of subjects, keywords, geographic location, and raw variables found on each dataset. At the same time, we performed a sanity check on each dataset and verified that the datasets are indeed a reliable gold standard.

Finally, the gold standard will be used for assessing the accuracy and performance evaluation in order to analyze each of the datasets when using Stanford NER tool, the post-processing method, and LIMES tool.

## 6.2 Evaluation of String Similarity Measures

String similarity measure evaluation aims to determine the correct measure given by LIMES for each pipeline. As explained in Figure 5.2, BigDataOcean Harmonization Tool has two main pipelines, in which the first pipeline is divided into three sub-pipelines for subject, keywords and geographic location respectively, and the second pipeline is for variables. Additionally, in the section 5.2.2.2, it is stated that LIMES provides various types of string similarity to help correctly identify the links between the raw data and the terms/instances given in each vocabulary.

For this purpose, the evaluation assessed five string similarity measures, such as, *cosine similarity*, *Jaccard*, *overlap*, *trigram*, and *Qgrams*. The evaluation took a list of random entities to identify the correct link for the first main pipeline. The entities evaluated are *arctic*, *in situ*, *north atlantic ocean*, *baltic*, *oceanographic data*, *baltic sea*, *hydrographic conditions*, *temperature*, *salinity*, and *ocean*.

ID	Title	Text length	Number of entities	Number of variables
Copernicus 1	Global Ocean 1/12 Physics Analysis and Forecast updated Daily	3533	20	14
Copernicus 2	Mediterranean Sea Waves Analysis and Forecast	9699	31	17
Copernicus 3	Arctic Ocean Physics Analysis and Forecast	3944	21	15
Copernicus 4	Baltic Sea Physics Analysis and Forecast	4044	19	9
Copernicus 5	Atlantic - European North West Shelf - Ocean Physics Analysis and Forecast	1597	16	7
Copernicus 6	Atlantic-Iberian Biscay Irish- Ocean Physics Analysis and Forecast	15584	25	9
NetCDF 1	weekly mean fields from Global Ocean Biogeochemistry Analysis	77	5	12
NetCDF 2	The GEBCO One Minute Grid - a continuous terrain model for oceans and land at one arc-minute intervals	217	5	3
NetCDF 3	Arctic- NRT in situ Observations	54	5	14
CSV	anek history 20180101T102300 20180806T131000	44	1	7
Excel	forecast nc	11	1	15

TABLE 6.1: Gold standard dataset

Table 6.2 presents the evaluation for the subject pipeline. The table shows that LINES only identifies one entity correctly *ocean* when used the *Qgrams* similarity string giving a threshold 0.75 while the other three string similarity does not identify any correct link. Therefore, BigDataOcean Harmonization Tool uses the *Qgrams* metric with an acceptance threshold of 0.74 and a review threshold of 0.7.



Words	Correct Links	Incorrect Links	Cosine	Overlap /Tri-gram	Qgrams
in situ	-	Topic categories in accordance with EN ISO 19115	0.25	0.2	-
observations	-	Location	-	-	0.23
north atlantic ocean	-	Oceans	-	-	0.15
baltic	-	Health	-	-	0.14
oceanographic data	-	Oceans	-	-	0.17
hydrographic conditions	-	Utilities / Communication	-	-	0.1
temperature	-	Structure	-	-	0.14
ocean	Oceans	-	-	-	0.75

TABLE 6.2: Subjects string similarity measure evaluation

Words	Correct Links	Incorrect Links	Cosine	Jaccard	Overlap /Tri-gram	Qgrams
in situ	in situ	-	1	1	1	1
north atlantic ocean	North Atlantic Ocean	-	1	1	1	1
baltic sea	-	sea	0.707	0.333	0.666	0.125
temperature	temperature	-	1	1	1	1
salinity	-	water salinity	0.707	0.333	0.666	0.5
ocean	ocean	-	1	1	1	1

TABLE 6.3: Keywords string similarity measure evaluation

For the keywords pipeline, table 6.3 shows that LIMES presents better linking results when using the four string similarity measures. Further, this happened because the keywords vocabulary has over 200 terms/instances while the subject vocabulary only has an approximation of 15 terms/instances. BigDataOcean Harmonization Tool uses the *cosine similarity* measure because it has better linking results as the others. Therefore, the acceptance threshold is 0.95, and the review threshold is 0.7.

For the geographic location pipeline, table 6.4 shows that only two entities were correctly identified with a threshold of 1.0 and as the same behavior of keywords pipeline, the BigDataOcean Harmonization Tool works with the *cosine similarity* as a result of the better linking threshold. Therefore, the acceptance threshold is 0.95, and the review threshold is 0.7.

Words	Correct Links	Incorrect Links	Cosine	Jaccard	Overlap /Tri-gram	Qgrams
arctic	-	Arctic Ocean	0.7071	0.333	0.666	0.4
north atlantic ocean	North Atlantic Ocean	-	1	1	1	1
baltic	-	Baltic Sea	0.7071	0.333	0.666	0.5
baltic sea	Baltic Sea	-	1	1	1	1
ocean	-	Arctic Ocean	0.7071	0.333	0.666	0.3

TABLE 6.4: Geographic location string similarity measure evaluation

For the second pipeline, the entities are random raw variables taken from some datasets, as *ship\_id*, *speed*, *lon*, *lat*, *course*, *heading*, *timestamp*, *psal*, *air\_pressure\_at\_sea\_level*, *wdir*, and *name*.

Table 6.5 shows that all the string similarity measures give the same correct linking results with a threshold of 1.0. Therefore, the BigDataOcean Harmonization Tool works with the *cosine similarity* measure giving the acceptance threshold of 0.98, and the review threshold of 0.95.

Words	Correct Links (name)	All string similarities
ship_id	ship_id	1
speed	speed	1
lon	lon	1
lat	lat	1
course	course	1
heading	heading	1
timestamp	timestamp, timecounter	1
psal	psal, pmsl, atms	1
air_pressure_at_sea_level	atmospheric_pressure_ on_the_sea_level	1
wdir	wdir, wdir2	1
name	name, name2	1

TABLE 6.5: Variables string similarity measure evaluation

## 6.3 Accuracy Evaluation

The primary objective of accuracy evaluation is to evaluate and analyze the efficiency of the BigDataOcean Harmonization Tool using Stanford Named Entity Recognizer and LIMES. This section is divided into two parts, the first part presents the behavior of using post-processing Stanford NER with the raw data given by the datasets, and the second part presents the accuracy evaluation of various datasets and what results returned when using Stanford NER and LIMES.

For each dataset evaluation, the LIMES cache memory was empty to avoid bias. As recalled in section 3.2.2.1, LIMES will save the instances extracted from the vocabulary and dataset in the cache to provide better efficiency and performance in the tool.

### 6.3.1 Accuracy Evaluation with or without Post-processing Method before Stanford NER

One of the problems the BigDataOcean Harmonization Tool found was that the Stanford NER returned combined entities into one word avoiding to find more than one possible link entities when executing LIMES tool, as explained in section 5.2.1.1, e.g., *North Atlantic Ocean*. Figure 6.2 shows the behavior of executing Stanford NER with and without the post-processing method. As the figure shown, the dataset *Copernicus 1* without post-processing shows 44 entities while with post-processing shows 68 entities. The same will happen with the other five datasets, although only two of them has one entity such as CSV and Excel datasets.

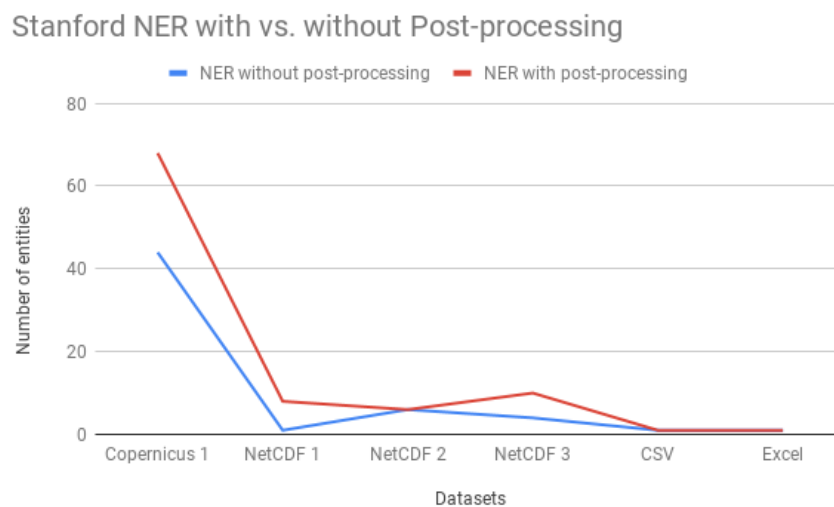


FIGURE 6.2: Stanford NER post-processing behavior

Using post-processing will give better results when the list is delivered to LINES. Table 6.6 shows what happens when dataset *Copernicus 1* is executed with and without the post-processing method before Stanford NER returns the list of entities, and which entities LINES linked. It is important to denote that the table only shows the matching entities which are linked by LINES. The left table is for entities without the post-processing method, as explained before the total is 44 entities but LINES only linked with 13 of them. On the other hand, the right table presents the entities with the post-processing method which has in total 68 entities but only 20 of them linked by LINES.

<i>(a) WITHOUT post-processing</i>	
ice	K
global ocean	G
model	K
energy	K
forecast	K
analysis	K
light	K
mediterranean sea	K, G
temperature	K
product	K
forecasting	K
time	K
sea level	K

<i>(b) WITH post-processing</i>	
weather	K
global ocean	G
model	K
energy	K
forecast	K
analysis	K
light	K
sea	K
temperature	K
product	K
in situ	K
forecasting	K
ice	K
ocean	S, K
mediterranean sea	K, G
satellite	K
time	K
sea level	K

TABLE 6.6: Linked results given by LINES using entities with and without post-processing (S = Subject, K = Keywords, G = Geographic Location)

Using the post-processing method is better considering that LINES identified 1 subject (S), 17 keywords (K), and 2 geographic locations (G) entities. While without the post-processing there are 12 keywords (K), 2 geographic locations (G), and no subject (S) entities returning bad efficiency results for the subject attribute.

### 6.3.2 Accuracy Evaluation with Post-processing Method using Stanford NER and LIMES

In this section, we will present the accuracy evaluation applying the gold standard dataset and using the evaluation measures, explained in section 2.3.2 when using Stanford NER tool, the post-processing method, and LIMES tool.

Table 6.7 presents the accuracy evaluation for the dataset *Copernicus 1*. The BigDataOcean Harmonization Tool returns many correct linked entities for each of the attributes given a result an F measure of 1.0.

Attributes	FP	FN	TP	P	R	F1
Subject	0	0	1	1	1	1
Keywords	0	0	17	1	1	1
Geographic Location	0	0	2	1	1	1
Variables	0	0	18	1	1	1

TABLE 6.7: Accuracy evaluation for the dataset “Copernicus 1”

The same behavior is displayed in the tables 6.8 and 6.9 where one is for the *NetCDF 3* and the other for *Excel* dataset respectively.

Attributes	FP	FN	TP	P	R	F1
Subject	0	0	1	1	1	1
Keywords	0	0	3	1	1	1
Geographic Location	0	0	1	1	1	1
Variables	0	0	19	1	1	1

TABLE 6.8: Accuracy evaluation for the dataset “NetCDF 3”

In the case of the table 6.9, only one entity has been correctly identified in the subject, keywords, and geographic location, the reason is that as seen in the table 6.1, the *Excel* dataset only has one entity which has recognized by the Stanford NER and the post-processing method.

Attributes	FP	FN	TP	P	R	F1
Subject	0	0	1	1	1	1
Keywords	0	0	1	1	1	1
Geographic Location	0	0	1	1	1	1
Variables	0	0	27	1	1	1

TABLE 6.9: Accuracy evaluation for the dataset “Excel”

Figures 6.3 and 6.4 present the behavior of the total number of entities returned by the Stanford NER and the post-processing method, also, the number of the subject, keywords, and geographic locations returned by LIMES. As seen in the figure, depending on the size of the entities list provided by the Stanford NER and the post-processing method, LIMES can link more accurate entity results to each pipeline (attribute).

Figure 6.3 presents the entities for only Copernicus datasets the reason is that only Copernicus datasets provide more information about the datasets than other formats. As seen in the figure depending on what description gave the better-linked results is given by LIMES. However, some datasets have no useful information which help to identify the attributes as happened with the *Copernicus 5* where it has no linked entities for geographic location.

Number of entities, Number of Subjects, Number of Keywords and Number of Geographic Locations only for Copernicus

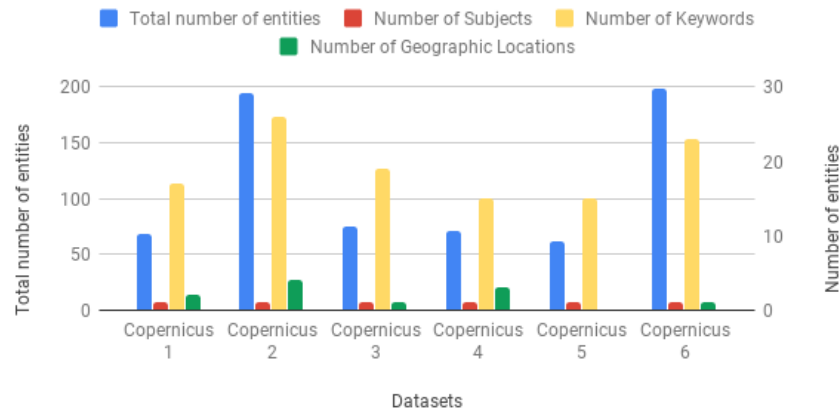


FIGURE 6.3: Total number of entities, number of subjects, number of keywords and number of geographic locations using Stanford NER and LIMES only for Copernicus datasets

On the other hand, when BigDataOcean Harmonization Tool executes another format different as Copernicus, the results can differ. Figure 6.4 presents the result entities for three NetCDF, one CSV, and one Excel, in this case, there is only two datasets *NetCDF 1* and *NetCDF 3* which have around 8 to 10 entities and each of them return some linked entities to each attribute. Nonetheless, the other three datasets have no linked entities for an attribute, e.g., *NetCDF 2* does not have entities for geographic location, *CSV* and *Excel* do not have linked entities for subject and geographic location.

Number of entities, Number of Subjects, Number of Keywords  
and Number of Geographic Locations

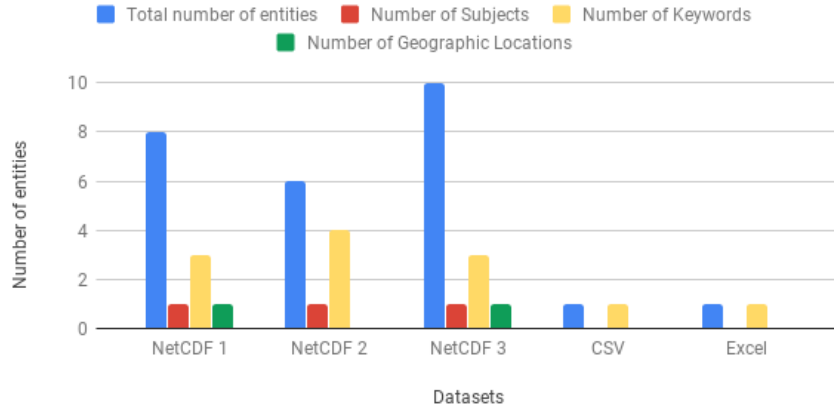


FIGURE 6.4: Total number of entities, number of subjects, number of keywords and number of geographic locations using Stanford NER and LIMES

In the case of the variables, the accuracy evaluation only depends on the raw variables provided by the dataset and the terms/instances provided by the Canonical Model Vocabulary, as explained in section 5.1. Figure 6.5 presents the linked results given by LIMES for the different formats. Here the raw variables can have more than one linked entity, therefore, for the *Excel* dataset there are 15 raw variables and 27 linked canonical variables, unlike, the *NetCDF 2* dataset has 3 raw variables which only has 1 entity for each raw variable given a total of 3 linked canonical variables. Table 6.10 presents an example of what is the results of the dataset *NetCDF 3* given by LIMES for the pipeline variables. As shown in the table, each raw variable has been mapped using the name or the canonical name of a canonical variable, for example, the raw variable *TIME* has been matched with the canonical name, and LIMES found out 3 links. Unlike the raw variable *CNDC* has been matched with the name. As well as, sometimes LIMES can match a raw variable with the name and the canonical name of a canonical variable, e.g., *LONGITUDE*.

We created a flexible pipeline evaluation, where we used the same string similarity measure for the subject, keywords and geographic location pipelines for the dataset *Copernicus 1*. For this evaluation, we used three measures, *Cosine similarity*, *Trigrams*, and *Qgrams* with an acceptance threshold of 0.8 for the keywords and geographic location pipelines, and 0.74 for the subject pipeline.

Table 6.11 shows the accuracy evaluation of using *Cosine similarity*, in this case, the F measure for subjects is 0.0. For *Trigrams* measure, table 6.12 shows that the F measure of the subject still is 0.0, but the keywords F measure is 0.94 due to LIMES identified two false positive (FP) link entities. Additionally, table 6.13 shows that using the *Qgrams*

Number of raw variables and Number of linked variables using LINES

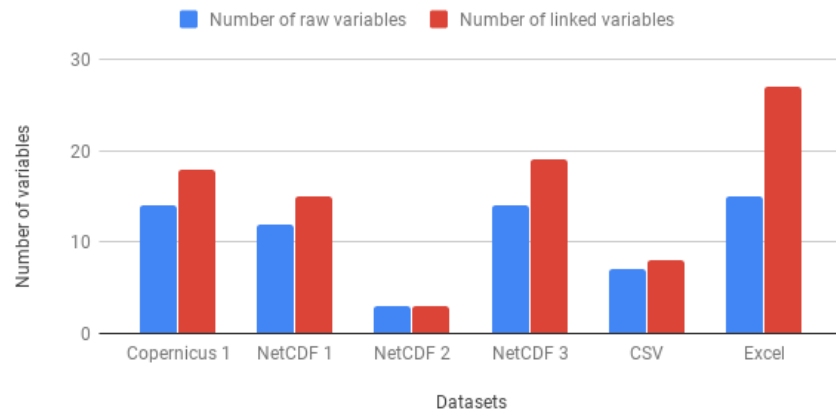


FIGURE 6.5: Number of raw variables and Number of linked variables using LINES

Raw variables	Name:CanonicalName
TIME	tgps:time, timestamp:time, time:time
TIME_QC	time_qc:time_quality_flag
LATITUDE	lat:latitude, latitude:latitude
LONGITUDE	longitude:longitude, lon:longitude
DIRECTION	direction1:direction, direction:direction
POSITION_QC	position_qc:position_quality_flag
DEPH	deph:manually_entered_depth
DEPH_QC	deph_qc:manually_entered_depth_quality_flag
CNDC	cndc:sea_water_electrical_conductivity
CNDC_QC	cndc_qc:sea_water_electrical_conductivity_quality_flag
TEMP	temp:sea_water_temperature
TEMP_QC	temp_qc:sea_water_temperature_quality_flag
PSAL	psal:sea_water_practical_salinity
PSAL_QC	psal_qc:sea_water_practical_salinity_quality_flag

TABLE 6.10: Canonical named variables for the dataset for the dataset “NetCDF 3” linked with raw variables using LINES

measure shows that the subject has now one linked entity giving an F measure of 1.0 and the other two pipelines show the same behavior as the *Cosine similarity* measure giving an F measure of 1.0.

Attributes	FP	FN	TP	P	R	F1
Subject	0	1	0	0	0	0
Keywords	0	0	17	1	1	1
Geographic Location	0	0	2	1	1	1

TABLE 6.11: Accuracy evaluation for the dataset “Copernicus 1” with Cosine similarity measure



Attributes	FP	FN	TP	P	R	F1
Subject	0	1	0	0	0	0
Keywords	2	0	17	0.84	1	0.94
Geographic Location	0	0	2	1	1	1

TABLE 6.12: Accuracy evaluation for the dataset “Copernicus 1” with Trigrams measure

Attributes	FP	FN	TP	P	R	F1
Subject	0	0	1	1	1	1
Keywords	0	0	17	1	1	1
Geographic Location	0	0	2	1	1	1

TABLE 6.13: Accuracy evaluation for the dataset “Copernicus 1” with Qgrams measure

## 6.4 Performance Evaluation

At the same time, the accuracy evaluation was carried out, BigDataOcean Harmonization Tool analyzed the performance evaluation. Here the assessment will show the time consumed when BigDataOcean Harmonization Tool received a dataset and extract the information needed to suggest the metadata using the Stanford NER tool, post-processing method, and LINES tool. The evaluation was performed on a laptop with Intel Corei7 4710HQ processor and 15GB RAM.

Figures 6.6 and 6.7 present the performance evaluation when a dataset is executed avoiding the pipeline execution of variables. In figure 6.6, the evaluation is for datasets which format is Copernicus, as seen in the section 6.3.2, Copernicus datasets provide more information making easy to find better-linked results with LINES, despite a large number of entities, the process is fast, giving less than six seconds to execute all the process giving at the end the suggested metadata to a dataset.

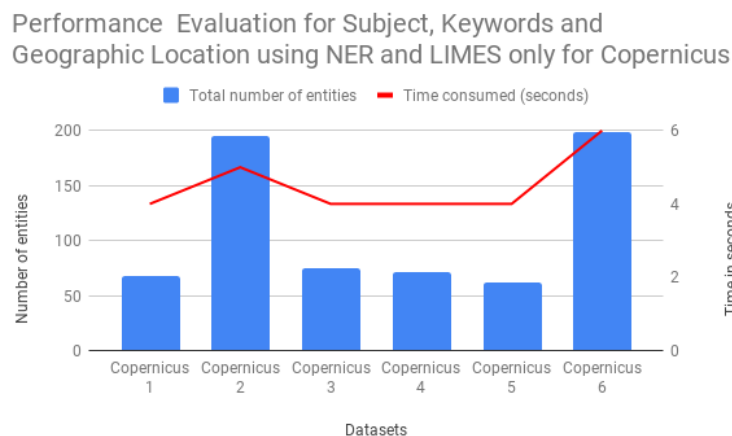


FIGURE 6.6: Performance evaluation for subject, keywords, and geographic location using Stanford NER and LINES only for Copernicus datasets

In the case of figure 6.7, the process time to suggest metadata to a dataset using Stanford NER and LINES is half the approximate time as in Copernicus datasets, knowing that there are fewer entities to extract using NER or LINES.

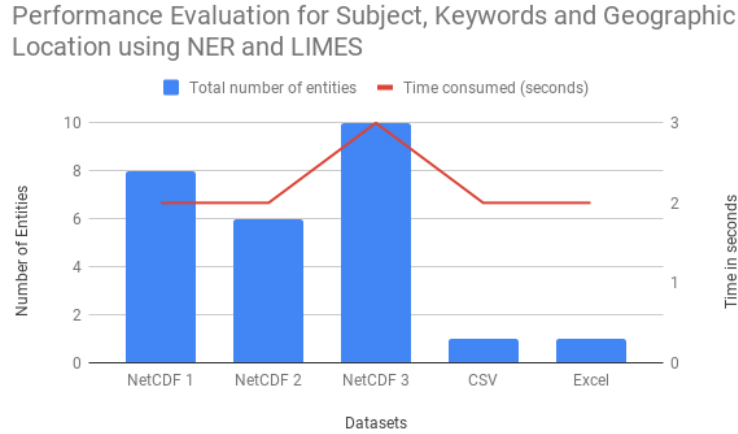


FIGURE 6.7: Performance evaluation for subject, keywords, and geographic location using Stanford NER and LINES

For the pipeline execution of variables is important to denote that the other pipelines were omitted only to analyze the performance evaluation of using only LINES tool. Figure 6.8 presents the execution time when a dataset provided to BigDataOcean Harmonization Tool to extract the information needed to run LINES and returns the corresponding canonical names of the raw variables. As seen in the figure, when there are less than 10 raw variables, then the process takes approximately one second. On the other hand, if the dataset has more than 20 raw variables, the process is not as slow as a result of the time only increase one second compared with raw variables below 10.

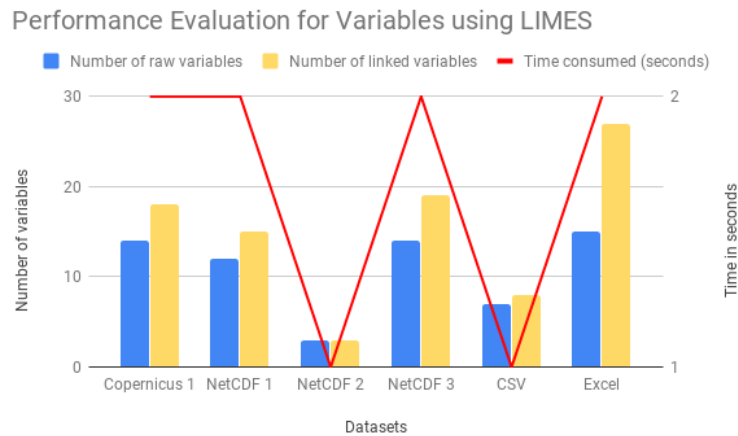


FIGURE 6.8: Performance evaluation for variables using LINES

In general, all the evaluations which the BigDataOcean Harmonization Tool took using the post-processing method helped to get better-linked results in each of the attributes (subject, keywords, geographic location).

## Chapter 7

# Conclusions

This thesis reported the state of the art of interlinking tools (LIMES and Silk) and natural language processing for named entity recognition tools (Stanford NER and Apache OpenNLP), the description of the BigDataOcean project, and the methodology implemented to extend the BigDataOcean project using Automated Link Discovery on the Semantic Web.

From state of the art, we conclude LIMES and the Stanford NER are better suited for Automated Link Discovery process given that LIMES is faster by reducing the number of comparisons needed during mapping using the caching method, and offers RESTful API. Stanford NER can extract all tags specified by a single annotator model, is state-of-the-art technology, and the accuracy and time performance is better than Apache OpenNLP.

Therefore, we applied LIMES and Stanford NER tools in the BigDataOcean Harmonization Tool to create the Automated Link Discovery process and evaluated the generated results.

In conclusion, we outline the implications of this thesis as follows:

- Interlinking BigDataOcean Harmonization Tool to vocabularies extracted from BigDataOcean Vocabulary Repository leads to an automatic enhancement of the dataset metadata.
- Using flexible string similarity measures in LIMES demonstrates that the F measure can variate giving bad results 0.0 or good results 1.0 for each pipeline presented in the thesis. Therefore, evaluating the string similarity measures demonstrates that applying Cosine Similarity and Qgrams help to better linking results performing an F measure of 1.0.

- Evaluating the results of named entity recognition and interlinking methods demonstrates that the accuracy of the post-processing method in the Stanford NER is higher than without the post-processing since we have achieved a better precision, recall and F measure.
- The evaluation of the performance of Stanford NER and LIMES for different size datasets shows that less time up to 6 seconds was spent presenting the dataset metadata.
- Using LIMES helps to get better and faster results when identifying the singular or multiple canonical variables for raw variables.

Until this point, BigDataOcean Harmonization Tool is capable of extracting/importing metadata of new datasets using the Automated Link Discovery process, as future work, it is intended that the BigDataOcean Harmonization Tool experiments by adding new vocabularies to be interlinked automatically with the subject pipeline, helping get at least more than one matching result. Also, the BigDataOcean Harmonization Tool can search for new interlinked results when the user has updated the title or the description. Finally, BigDataOcean Harmonization Tool can be extended to understand multi-lingual data sources.

# Bibliography

- [1] Linked Data. Linked data - connect distributed data across the web, 2018. URL <http://linkeddata.org/>.
- [2] Accuracy, precision, recall & f1 score: Interpretation of performance measures. <https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/>, 2016.
- [3] Linked data. <https://www.w3.org/standards/semanticweb/data>, 2015.
- [4] Ioanna Lytra, Fabrizio Orlandi, and Maria-Esther Vidal. BigDataOcean - Exploiting Oceans of Data for Maritime Applications, May 2017. URL <https://doi.org/10.5281/zenodo.815343>. <https://2017.eswc-conferences.org/program/eu-project-networking-session>.
- [5] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):34–43, 2001.
- [6] Semantic web ontology. <https://www.w3.org/standards/semanticweb/ontology>, 2015.
- [7] Contolled vocabulary vs ontology. <https://semwebtec.wordpress.com/2010/11/23/contolled-vocabulary-vs-ontology/>, 2015.
- [8] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227. IGI Global, 2011.
- [9] Enayat Rajabi, Miguel-Angel Sicilia, and Salvador Sanchez-Alonso. An empirical study on the evaluation of interlinking tools on the web of data. *Journal of Information Science*, 40(5):637–648, 2014.
- [10] Samet Atdağ and Vincent Labatut. A comparison of named entity recognition tools applied to biographical texts. In *Systems and Computer Science (ICSCS), 2013 2nd International Conference on*, pages 228–233. IEEE, 2013.

- [11] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *International Journal of Machine Learning Technology*, 2011.
- [12] Yutaka Sasaki et al. The truth of the f-measure. *Teach Tutor mater*, 1(5):1–5, 2007.
- [13] Albert Weichselbraun, Daniel Streiff, and Arno Scharl. Consolidating heterogeneous enterprise data for named entity linking and web intelligence. *International Journal on Artificial Intelligence Tools*, 24(2):1540008, 2015.
- [14] Yves Raimond, Christopher Sutton, and Mark B Sandler. Automatic interlinking of music datasets on the semantic web. *LDOW*, 369, 2008.
- [15] Muhammad Saleem, Shanmukha S Padmanabhuni, Axel-Cyrille Ngonga Ngomo, Jonas S Almeida, Stefan Decker, and Helena F Deus. Linked cancer genome atlas database. In *Proceedings of the 9th International Conference on Semantic Systems*, pages 129–134. ACM, 2013.
- [16] Oktie Hassanzadeh and Mariano P Consens. Linked movie data base. In *LDOW*, 2009.
- [17] Alexander Arturo Mera Caraballo, Bernardo Pereira Nunes, and Marco A Casanova. Drx: A lod browser and dataset interlinking recommendation tool. *Semantic Web Journal*, 2009.
- [18] Ying Zhang, Yao-Yi Chiang, Pedro Szekely, and Craig A Knoblock. A semantic approach to retrieving, linking, and integrating heterogeneous geospatial data. In *Joint Proceedings of the Workshop on AI Problems and Approaches for Intelligent Environments and Workshop on Semantic Cities*, pages 31–37. ACM, 2013.
- [19] Nelson Piedra and Juan Pablo Suárez. Smartland-ld: A linked data approach for integration of heterogeneous datasets to intelligent management of high biodiversity territories. In *International Conference on Software Process Improvement*, pages 207–218. Springer, 2017.
- [20] Gobinda G Chowdhury. Natural language processing. *Annual review of information science and technology*, 37(1):51–89, 2003.
- [21] Andrei Mikheev, Marc Moens, and Claire Grover. Named entity recognition without gazetteers. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 1–8. Association for Computational Linguistics, 1999.

- [22] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.
- [23] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, 2007.
- [24] David Campos, Sérgio Matos, and José Luís Oliveira. Biomedical named entity recognition: a survey of machine-learning tools. In *Theory and Applications for Advanced Text Mining*. InTech, 2012.
- [25] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *arXiv preprint arXiv:1812.09449*, 2018.
- [26] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics, 2005.
- [27] Grant S Ingersoll, Thomas S Morton, and Andrew L Farris. *Taming text: how to find, organize, and manipulate it*. Manning Publications Co., 2013.
- [28] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.
- [29] Stephan Wölger, Katharina Siorpaes, Tobias Bürger, Elena Simperl, Stefan Thaler, and Christian Hofer. A survey on data interlinking methods, 2011.
- [30] Alfio Ferrara, Andriy Nikolov, and François Scharffe. Data linking for the semantic web. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 7(3):46–76, 2011.
- [31] Elena Simperl, Stephan Wölger, Stefan Thaler, Barry Norton, and Tobias Bürger. Combining human and computation intelligence: the case of data interlinking tools. *International Journal of Metadata, Semantics and Ontologies*, 7(2):77–92, 2012.
- [32] Markus Nentwig, Michael Hartung, Axel-Cyrille Ngonga Ngomo, and Erhard Rahm. A survey of current link discovery frameworks. *Semantic Web*, 8(3):419–436, 2017.
- [33] Axel-Cyrille Ngonga Ngomo and Sören Auer. Limes-a time-efficient approach for large-scale link discovery on the web of data. In *IJCAI*, pages 2312–2317, 2011.
- [34] LIMES. *LIMES Documentation: User Manual*. AKSW, 2018.



- [35] Wael H Gomaa and Aly A Fahmy. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18, 2013.
- [36] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Silk-a link discovery framework for the web of data. *LDOW*, 538, 2009.
- [37] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Discovering and maintaining links on the web of data. In *International Semantic Web Conference*, pages 650–665. Springer, 2009.
- [38] Ana Cristina Trillos Ujueta. Survey on metadata repositories for vocabularies and ontologies. Master’s thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 6 2018.
- [39] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.
- [40] Jiaheng Lu, Chunbin Lin, Wei Wang, Chen Li, and Haiyong Wang. String similarity measures and joins with synonyms. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 373–384. ACM, 2013.