A FRAMEWORK FOR ANALYSING THE COMPLEXITY OF ONTOLOGY

by

Yannick Kazela Kazadi

210122927

Dissertation submitted in fulfilment of the requirement for the degree of Magister Technologiae: Information Technology

in the

Department of Information and Communication Technology, Faculty of Applied and Computer Sciences, Vaal University of Technology

Supervisor: Dr J.V. Fonou-Dombeu

Co-Supervisor: Prof. K. Okosun

November 2016

DECLARATION

I hereby declare that this dissertation, which I submit for the qualification of

Magister Technologiae: Information Technology

To the Vaal University of Technology, Department of Information and Communication Technology, Faculty of Applied and Computer Sciences, apart from the recognized assistance of my supervisor, co-supervisor and provided citations, is my own work and has not previously been submitted to any other institution for any degree.

_________________________________ On this _____________ day of ______________

Candidate

_________________________________On this______________day of ______________

Supervisor

DEDICATION

I dedicate and deeply thank God for supplying me with all the necessary resources I needed for me to complete this work, my mother Anny who has done so many sacrifices for me, my father Pierre for being an inspiration for me, my wife Rachel who has supported and motivated for completing this work, and finally my sisters and brothers Sandra, Gracia, Prisca, Hodavia, Perfetia Ben and Winner, my niece Candela and my grandmother Esther.

ACKNOWLEDGEMENTS

I deeply thank and express my gratitude to my supervisor Dr J.V. Fonou-Dombeu for his support which has been far beyond his role of supervisor. I also address my gratitude to my family for believing in me and helping me to believe in myself in difficult moments. I would like to thank people who directly or indirectly participated to my education.

PUBLICATIONS

KAZADI, Y.K. & FONOU-DOMBEU, J.V. Adaptive Algorithms for Computing Ontologies Metrics through Processing of RDF Graphs, *International Journal of Semantic Web and Information Systems (IJSWIS)*, Submitted.

KAZADI, Y.K. & FONOU-DOMBEU, J.V. (2016a) Analysis of Advanced Complexity Metrics of Biomedical Ontologies in the BioPortal Repository, *In Proceedings of the 1st International Conference on Complex Information Systems (COMPLEXIS 2016)*, Rome, Italy, ISBN: 978-989-758-181-6, 22-24 April, pp. 107-104.

KAZADI, Y.K. & FONOU-DOMBEU, J.V. (2016b) Complexity Based Ranking of Biomedical Ontologies, *In Proceedings of the 3rd IEEE International Conference on Advances in Computing, Communication & Engineering 2016 (ICACCE 2016)*, Durban, South Africa, ISBN: 987-1-5090-2576-6, 28-29 November, pp. 423-429.

ABSTRACT

The emergence of the Semantic Web has resulted in more and more large-scale ontologies being developed in real-world applications to represent and integrate knowledge and data in various domains. This has given rise to the problem of selection of the appropriate ontology for reuse, among the set of ontologies describing a domain. To address such problem, it is argued that the evaluation of the complexity of ontologies of a domain can assist in determining the suitable ontologies for the purpose of reuse. This study investigates existing metrics for measuring the design complexity of ontologies and implements these metrics in a framework that provides a stepwise process for evaluating the complexity of ontologies of a knowledge domain. The implementation of the framework goes through a certain number of phases including the: (1) download of 100 Biomedical ontologies from the BioPortal repository to constitute the dataset, (2) the design of a set of algorithms to compute the complexity metrics of the ontologies in the dataset including the depth of inheritance (DIP), size of the vocabulary (SOV), entropy of ontology graphs (EOG), average part length (APL) and average number of paths per class (ANP), the tree impurity (TIP), relationship richness (RR) and class richness (CR), (3) ranking of the ontologies in the dataset through the aggregation of their complexity metrics using 5 Multi-attributes Decision Making (MADM) methods, namely, Weighted Sum Method (WSM), Weighted Product Method (WPM), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), Weighted Linear Combination Ranking Technique (WLCRT) and Elimination and Choice Translating Reality (ELECTRE) and (4) validation of the framework through the summary of the results of the previous phases and analysis of their impact on the issues of selection and reuse of the biomedical ontologies in the dataset. The ranking results of the study constitute important guidelines for the selection and reuse of biomedical ontologies in the dataset. Although the proposed framework in this study has been applied in the biomedical domain, it could be applied in any other domain of Semantic Web to analyze the complexity of ontologies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1. BACKGROUND

Ontology is a formal, explicit specification of a shared conceptualisation of a domain of knowledge (Gruber 1993). It represents knowledge as a set of concepts within a domain, and the relationships between pairs of concepts. Ontologies constitute the backbone of Semantic Web applications; they have become a key technology in providing shared knowledge models to semantic-driven applications (Yang, Zhang & Ye 2006).

As ontologies grow in size and number, it is important to evaluate their complexity quantitatively to enable developers to better understand, maintain, reuse and integrate them (Zhang, Li & Tan 2010). Yang et al. (2006) added that ontology evaluation enables developers to determine the fundamental characteristics of ontologies in order to improve the quality, estimate cost and reduce future maintenance.

Bontas, Mochol and Tolksdorf (2006) defined ontology reuse as the process in which available ontological knowledge is used as the input to generate new ontologies. Evaluating the complexity of ontologies can, therefore, help determine which ontology to select and submit to the process of reuse; it can also provide a better way for selecting the necessary knowledge from the chosen ontology. Furthermore, a quantitative measurement of the complexity of ontology can improve the understanding of the structure of ontology and its semantics to developers of ontologies, thereby allowing them to better evaluate ontology design and control its development process (Zhang et al. 2010).

1.2. RATIONAL AND MOTIVATION

With the rising importance of knowledge exchange on the World Wide Web, ontologies have become a key technology in providing shared knowledge to semantic-driven applications (Yang et al. 2006). Brank, Grobelnik and Mladenic (2005) argued that ontology evaluation is an important issue that must be addressed if ontologies are to be widely utilized in the Semantic Web and other semantics-aware applications. As any other resource used in software applications, ontologies need to be evaluated before their use to prevent applications from using inconsistent, incorrect, or redundant ontologies (Cross & Pal 2008). Furthermore, an evaluation and analysis of ontology can assist developers in selecting from the large number of available ontologies in a domain, the appropriate candidate ontologies for their application (Natalya 2004; Brank et al. 2005; Cross & Pal 2008). In fact, a well-defined, designed and built ontology greatly determines the quality of the application that uses it as a source of data and a means for organizing knowledge of a given domain. Moreover, evaluating and analysing the quality of ontologies of a domain strengthen their reuse and reduce the time and effort required to build and maintain new ontologies on the same domain (Brewster, Alani, Dasmapatra & Wilks 2004; Zhang et al. 2010).

1.3. PROBLEM STATEMENT

There are very few commonly agreed methodologies and metrics for analysing the complexity of ontology and evaluating its evolution (Stojanovic & Motik 2002). According to Zhang et al. (2010), the emergence of the Semantic Web has resulted in more and more large-scale ontologies being developed in real-world applications to represent and integrate knowledge and data in various domains of knowledge.

This raises the problem of selecting, among a set of ontologies describing a domain, one or more that can satisfy the requirements of developers and users. The measurement of the complexity of ontologies can constitute one effective criterion for making an accurate choice. As specified by Zhang et al. (2010) the quantitative measurement of complexity can help ontology developers and maintainers to understand the current status of the ontology, evaluate its design and control its development process.

This study investigates existing metrics used to measure the design complexity of ontologies and implement them in a framework that provides a stepwise process for evaluating the complexity of ontologies. The proposed framework would be useful to Semantic Web developers in that it would enable them to choose among the set of ontologies describing a domain based on their levels of complexity; the framework would also foster ontology reuse across Semantic Web application domains (Bontas et al. 2006; Pak & Zhou 2011).

1.4. AIM AND RESEARCH OBJECTIVES

The aim of this research is to design and apply a framework for analysing the complexity of ontologies. The objectives of this study are:

1. To review existing metrics for measuring the complexity of ontologies.
2. To acquire the ontologies of a selected domain of knowledge on the Semantic Web.
3. To develop a framework for analysing the complexity of ontologies of a given domain based on existing ontology complexity metrics.

.

1.5. RESEARCH METHODOLOGY

1.5.1 Data Collection

Data collection was done by a literature search. Journal articles, conference papers and books related to the topics of ontology complexity analysis were targeted.

1.5.2 Framework Design

The proposed framework constituted three main components:

- **ontology store and management** – stores ontologies that are to be evaluated, in RDF (Resource Description Framework) or OWL (Web Ontology Language) formats using existing ontologies storage and query toolkits (Ramanujam, Gupta, Khan, Seida & Thuraisingham 2009; Fan, Zhang, Zhao 2010; Zhou 2010) and,
- **ontology analysis and evaluation** – Implements algorithms for ontology complexity analysis (Lozano-Tello, Gomez-Perez & Sosa 2004; Tartir, Arpinar, Moore, Sheth & Aleman-Meza 2005; Alani & Brewster 2006).
- **Ontology ranking** – Implements existing ranking algorithms (Benayoun et al. 1966; Fishburn 1967; Miller et al. 1969; Saaty 1977; Chou 2013) to rank the ontologies based on the aggregation of their complexity metrics.

1.5.3 Experiments

The experiments in this study were carried out using the following tasks.

- **Ontology Search** - ontologies of a chosen domain were searched and selected from the World Wide Web by means of Semantic Web ontology search engines including Swoogle, Watson, SWSE, Powerset, Kngine, etc. as well as ontologies repositories (Rodriguez, Sicilia & Garcia 2012; Sudeepthi, Anuradha & Babu 2012).

- **Ontology storage** – the selected ontologies were stored in existing ontology storage and query toolkits such as Jena API, AllegroGraph, Sesame, Minerva (Ramanujam et al. 2009; Fan et al. 2010; Zhou 2010).
- **Ontology analysis and Ranking** – algorithms were written in Eclipse Java IDE (Integrated Development Environment) configured with Semantic Web API (Application Programming Interface) such as Jena to analyse, evaluate and rank the ontologies based on their complexity.

1.5.4 Performance Evaluation

The performances of the framework were evaluated based on the following criteria:

- **Complexity metrics** – analysis of ontology complexity metrics to determine the levels of complexity of ontologies of a domain of knowledge,
- **Performance of algorithms** - evaluation of the execution times of algorithms for computing ontologies complexity metrics against the size of ontologies in the chosen domain of knowledge and
- **Ranking results** – analysis of the ranking results of algorithms and their impact on complexity of ontologies of the chosen domain of knowledge.

1.6. DISSERTATION OUTLINE

Chapter 2 presents the background on ontology evaluation and the *state-of-the-art* of ontology complexity analysis. In Chapter 3, ontology acquisition, algorithms for computing ontology complexity metrics and ranking of ontologies based on these metrics are discussed. Chapter 4 presents the proposed framework for analysing ontology complexity. Experimental results and discussions are presented in Chapter 5. A conclusion and future work is presented in Chapter 6.

1.7. ORIGINAL CONTRIBUTIONS

The original contributions made by this study are as follows:

1. In Chapter 3 Section 3.3, a set of algorithms for computing ontology complexity metrics through the processing of RDF graphs was developed. This original work was

submitted for review to *International Journal of Semantic Web and Information Systems (IJSWIS)*.

2. In Chapter 5 Section 5.4, the complexity metrics of 100 ontologies of the biomedical domain are computed and analysed to assess the level of complexity of these ontologies. This work was published in Kazadi & Fonou-Dombeu (2016a).
3. In Chapter 4 Section 4.2, a framework for the analysis of ontology complexity is proposed. This framework is constituted of 4 phases, namely, ontology acquisition, complexity metric computation, ontology ranking and validation. The results of the experimental evaluation of this framework were published in Kazadi & Fonou-Dombeu (2016b).

1.8 PUBLICATIONS

The following publications have resulted from this work:

KAZADI, Y.K. & FONOU-DOMBEU, J.V. Adaptive Algorithms for Computing Ontologies Metrics through Processing of RDF Graphs, *International Journal of Semantic Web and Information Systems (IJSWIS)*, Submitted.

KAZADI, Y.K. & FONOU-DOMBEU, J.V. (2016a) Analysis of Advanced Complexity Metrics of Biomedical Ontologies in the BioPortal Repository, *In Proceedings of the 1st International Conference on Complex Information Systems (COMPLEXIS 2016)*, Rome, Italy, ISBN: 978-989-758-181-6, 22-24 April, pp. 107-104.

KAZADI, Y.K. & FONOU-DOMBEU, J.V. (2016b) Complexity Based Ranking of Biomedical Ontologies, *In Proceedings of the 3rd IEEE International Conference on Advances in Computing, Communication & Engineering 2016 (ICACCE 2016),* Durban, South Africa, ISBN: 987-1-5090-2576-6, 28-29, November, pp. 423-429.

CHAPTER 2 :LITERATURE REVIEW

2.1 INTRODUCTION

As ontologies grow in size and number, it is important to evaluate and determine their complexity to better understand, maintain, reuse and integrate them (Zhang, Li & Tan 2010). It is argued that the growing demand for facilitating the deployment and reuse of ontologies has increased the need to develop adequate criteria to measure the quality of ontologies that conceptualize a domain (Supekar, Lee & Park 2004). This chapter therefore, intends to: (1) provide an overview of ontologies and their related components, (2) review the *state-of-the-art* in ontologies evaluation and analysis, and (3) presents and describes different methodologies and metrics proposed in the literature for the evaluation of ontology complexity.

2.2 ONTOLOGY

2.2.1 Definition of Ontology

Ontology is a formal, explicit specification of a shared conceptualisation of a domain of knowledge (Gruber 1993). It represents knowledge as a set of concepts within a domain, and the relations between pairs of concepts. According to Neches, Fikes, Finin, Gruber, Senator & Swartout (1990), an ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms. Ontology can be seen as an effective way to represent knowledge of a specific domain such as biomedicine, education, e-government, etc.

2.2.2 Ontology Components

An ontology is composed mainly of:

- **Concepts** - A material object, a notion or an idea (Ushold & King 1995). A concept is characterised by three elements: (1) the term of the concept, it is used to designate the concept, (2) the semantic of the concept comprises of the attributes and properties of the concept and (3) the instances of the concept.
- **Property –** relationship between concepts; it can be of two types: sub-class-of (generalisation/specification) and part-of (aggregation/composition) (Pal 2005).

- **Axiom –** Assertion or statement of an ontology; it is written in formal language like Resource Description Framework (RDF) or Web Ontology Language (OWL).
- **Restriction –** Condition set on ontology concepts, properties, and instances within an axiom.
- **Rule** – Logical statement that infers knowledge from an axiom (Guizzardi, Falbo, Pereira & Filho 2002; Kalibatiene,Vasilecas & Guizzardi 2007).
- **Inheritance hierarchy of ontology** - Sub-graph of this ontology's graph composed of classes and subclass relationships (Tartir, Arpinar, Moore, Sheth & Aleman-Meza 2005; Zhang et al. 2010).

2.2.3 Classification of Ontologies

There are many classifications of ontologies in the literature (Van Heijst, Schreiber & Wielinga 1997; Lassila and McGuinness 2001; Borgo 2007, etc). Thus, ontologies vary not only in their content but also in their structure and domains they represent (Pal 2005). In Van Heijst et al. (1997), ontologies are classified according to two dimensions: the amount and type of structure of the conceptualization and the subject of the conceptualization. This classification is similar to the one adopted by Roussey, Pinet, Kang & Corcho (2011) where ontologies are classified based on two criteria: the expressivity and formality of the languages used and the scope of the objects described. The categories of ontologies in the first classification are:

- **Information ontology -** Record the structured information such as database schemata (Rector, Nowlan, Kay, Goble & Howkins 1993).
- **Linguistic/Terminological ontology -** Represent natural languages and can be glossaries, dictionaries, controlled vocabularies, taxonomies, folksonomies, or lexical databases. It has two purposes: define the vocabulary and represent agreements between a users' community (Lindberg, Humphreys & McCray 1993).
- **Software ontology -** Conceptual representation of a domain with the aim of storing and manipulating data; it is used for software development activities with the goal of guaranteeing data consistency (Roussey et al. 2011).
- **Formal ontology -** Ontology represented in a formal language such as RDF(S) or OWL (Borgo 2004).

With respect to the scope of the objects/subjects described, ontologies are classified in the following categories:

- **Domain ontology -** Conceptualization that is specific to a particular domain; it is an ontology that is applicable only to a specific domain (Roussey et al 2011).
- **Local ontologies/application ontology -** Specializations of domain ontologies with no consensus or knowledge sharing (Tu, Eriksson, Gennari, Sharar & Musen 1995).
- **Core reference ontology –** Also called generic ontology, it is a standard used by different group of users. This type of ontology is linked to a domain but it integrates different viewpoints related to specific group of users (Heijst et al. 1997).
- **General ontology -** Represents general knowledge of a wide area; it is intended to be neutral with respect to world entities (Guarino & Boldrin 1993).

2.3 EVALUATION AND ANALYSIS OF ONTOLOGIES

Ontologies are useful for representing and conceptualizing a domain of knowledge. However, there is a great variation in the quality of ontologies representing the same domain of knowledge (Orbrst, Ashpole, Ceusters, Mani, Ray & Smith 2006). Therefore, for ontologies to be widely adopted by users, enterprises and communities there should be appropriate methodologies that enable the assessment of the quality of ontologies describing a domain as well as their ranking to provide guidelines for their reuse.

2.3.1 Approaches of Ontologies Evaluation

Different approaches for evaluating and analysing ontologies have been proposed (Brank, Grobelnik & Mladenic 2004; Ben Abbes 2009). These approaches include:

- **Use of a golden standard-** The ontology is compared to a golden standard which may also be an ontology (Bank et al. 2004). An example of the application of this approach is proposed by Maedche & Staab (2002). They determined the level of similarity of two ontologies by calculating the lexical similarity (LM) between two terms from both ontologies and deduced the similarity of two ontologies as the average of the different LM. The lexical content of an ontology can also be evaluated using the notions of Precision and Recall (Brewster et al. 2004; Velardi et al. 2005). Precision is defined as the percentage of the ontology lexical entries (strings used as concept) that also appear in the golden standard, relative to the total number of ontology concepts; whereas Recall is the percentage of the golden standard lexical entries that also appear as concept in the ontology, relative to the total number of golden standard lexical entries.

- **Using the ontology in an application and evaluating the results -** With this approach, ontologies are evaluated simply by plugging them into an application and evaluating the outputs of the application. According to Brank et al. (2004), this is elegant in the sense that the outputs of the application might be something for which a relatively straightforward and non-problematic evaluation approach already exists. In their work, Porzel & Malaka (2005) proposed an application of this approach and highlighted its disadvantages as follows:

    - Difficulty to generalize the results obtained since they depend only on the application used for the evaluation,
    - The ontology could be only a small component of the application and its effect on the outcome may be relatively small and indirect and
    - Comparing different ontologies is only possible if they can all be plugged into the same application.

- **Human assessments under specific criteria or standards**

The aim of this approach is to apply several criteria in order to determine the "good ontology" that complies with these criteria. For each criterion, the ontology is evaluated and given a numerical score. An overall score for the ontology is then computed as a weighted sum of its per-criterion scores. Gangemi, Catenacci, Ciaramita & Lehman (2005) classify these criteria into three different categories: structural, functional and usability-related.

The structural criteria are applied on ontologies represented as graphs. In this form, the topological and logical properties of an ontology may be measured by means of metrics. The existence of these structural dimensions, however, can be considered independent from the metric being used. Metrics are used to evaluate hierarchical relations between concepts in an ontology. Structural criteria comprise measures such as: the width, the depth, leaves distribution, cohesion, etc. (Gangemi et al. 2005). The functional criteria evaluate the intended use of a given ontology and of its components, i.e., its function. Functional dimensions include agreement, task, topic, design, etc. Such dimensions become apparent in an ontology depending on the context defined by how the ontology is chosen, built, exploited (Gangemi et al. 2005).

- **Data or corpus driven approach**

An ontology may also be evaluated by comparing it to existing data such as a collection of textual documents about the domain it represents. Patel, Supekar, Lee & Park (2003) provided an application of this approach to extract textual data from the ontology such as names of concepts and relations and used this as the input to a text classification model trained using standard machine learning algorithms. Another application in Brewster et al. (2004) extracted a set of relevant domain-specific terms from the corpus of documents using latent semantic analysis. The amount of overlap between the domain-specific terms and the terms appearing in the ontology were then used to determine how the ontology matches with the corpus.

2.3.2 Frameworks for Ontology Evaluation

In recent years a great effort has been invested into developing methodologies and tools that can be used to evaluate ontologies (Lethbridge 1998). This subsection looks at the existing metric-based frameworks for ontology evaluation. These metrics are either used to analyse the structure or evaluate the content of an ontology (Gangemi et al. 2005; Esposito, Zappatore & Tarricone 2011). The metrics used to analyse the size and structural aspects of an ontology can be implemented in automated or semi-automated tools, while those used to assess the ontology content focus on analysing the semantic meaning of ontology components and often require a domain expert contribution (Esposito et al 2011).

2.3.2.1 OntoMetric

OntoMetric is an ontology evaluation tool proposed by Lozano-Tello & Gómez-Pérez (2004). It is a hierarchical framework that consists of 160 characteristics called the multilevel framework of characteristics and spread across five dimensions to evaluate the quality and suitability of ontologies to users' system requirements. Knowledge engineers need to examine the characteristics of the ontologies in order to determine the best ontology for reuse in their system; this is done by supplying the application with several values that will be used to measure the suitability of an ontology for some given system's requirements. Dimensions defined in OntoMetric are: the content and organization of the ontology, the language in which it is implemented, the methodology that has been followed to develop it, the software tools used to build and edit the ontology, and the cost which refers to the amount of time and infrastructure needed to develop the ontology. Each dimension has factors that are the fundamental elements to be analysed in order to obtain the aggregated value for the dimension. Each factor is further subdivided into detailed characteristics.

.

2.3.2.2 OntoQA

Developed by Tartir, Arpinar, Moore, Sheth & Aleman-Meza (2005) at the University of Georgia, OntoQA is an ontology evaluation method and tool that analyses ontology schemas and their constituents (instances of concepts and properties) and describes them through a well-defined set of metrics. According to the authors of this method the quality of ontologies can be assessed in different dimensions. For example, quality metrics can be used to evaluate the success of a schema in modelling a real-world domain. The depth, breadth, and height balance of the schema inheritance tree can play a role in a quality assessment as well. The depth of inheritance of a given concept is the longest path from this concept to the root concept in the inheritance hierarchy of the ontology, whereas, the depth of inheritance of the ontology corresponds to the longest depth of inheritance of concepts of the ontology (Tartir et al. 2005). The height of a given concept, represented by a node in the ontology graph, is the longest path length from a leaf node to the node representing this concept (Dameron, Bettembourg & Le Meur 2013). The breadth of an ontology is defined as the number of levels of the inheritance tree of the ontology; it helps determine at what level the relevant concepts of the domain are covered by the ontology (Anand, Duin, Tavasszy & Wigan 2014).

In OntoQA, ontology metrics are divided into two related categories: schema metrics and knowledge-base (instance) metrics. The first category evaluates ontology design and its potential for rich knowledge representation. The second category evaluates the placement of data instances within the ontology and the effective utilization of the knowledge modelled in the schema.

2.3.2.3 OQuaRE

Proposed by Duque-Ramos, Fernández-Breis, Stevens & Aussenac-Gilles (2011), the OQuaRE is a framework for evaluating the quality of ontologies based on the SQuaRE standard for software quality evaluation (ISO25000 2005). This method requires the definition of both a quality model and quality metrics for evaluating the quality of the ontology. The quality model is divided into a series of quality dimensions or characteristics that are:

- **Structure** - ontological properties that are widely used in *state-the-of-art* evaluation approaches such as formalisation, formal relations support, cohesion, tangledness, redundancy and consistency.
- **Reliability** - capability of an ontology to maintain its level of performance under stated conditions for a given period of time.
- **Operability** - effort needed to use an ontology and the individual assessment of such use by a stated or implied set of users.
- **Maintainability -** the capability of ontologies to be modified to adapt to the changes in environments, requirements or functional specifications.

According to the developers of this framework, characteristics applicable to the evaluation of software can also be used to evaluate ontologies as they are software artefacts. The assessment of the maintainability and structural characteristics of an ontology are relevant as the effort of maintaining or bringing modifications to an ontology depends on how complex it is (Zhang et al. 2010).

2.3.2.4 OntoKhoj

Supekar et al. (2004) proposed a model for evaluating ontology schemas. The model contains two sets of features: quantifiable and non-quantifiable. Their technique is based on crawling the web to search for ontologies and store them locally, and then use information provided by the user, like the domain and weights of proposed metrics to return the most suitable ontology. The non-quantifiable features comprise the Cognitive Adequacy, that is, how well the ontology reflects the domain of interest and the Context which refers to the background information about the domain of interest that an agent needs to know before it seeks knowledge about the domain. The quantifiable features are:

- **Veracity** - a measure of correctness of the ontology with respect to the domain of discourse; it allows determining the syntactical errors that can be contained in an ontology code.
- **Practical quality of ontology** - how easy the ontology can be understood; this is essentially related to the complexity of its conceptualization. A more complex model would be difficult to comprehend by a user, thus, decreasing the usability of the ontology.

- **Complexity** - this feature in OntoKhoj is represented by a set of measures which are the depth and width of the ontology, its total number of relations, attributes and instances. The width of the ontology defines the number of children per concept.
- **Specificity** - this feature is used to determine how close an ontology is to the domain of discourse and it is also governed by the width and depth as a more specific ontology is indicative of high quality of knowledge.

2.4 STATISTICAL MEASUREMENT OF ONTOLOGY COMPLEXITY

2.4.1 Concept-Based Measurement of Ontology Complexity

Kang, Xu, Lu & Chu (2004) proposed a method that consists of weighting class dependence graphs to represent ontology and present a structured complexity measure of the ontology based on entropy distance. They consider the complexity of both the classes and relationships between the classes and present rules for transforming complexity value of classes along with different kinds of relations into a weighted class dependence graphs. Practically, weighting a class dependence graph consists in determining the complexity value of each class of the ontology graph and applying transformation rules to determine the weight of each relation of the ontology. The weight of a relation is calculated based on the complexity values of both classes involved in this relation.

In Yang et al. (2006) a suite of metrics for the measurement of the complexity of ontology is suggested. These metrics mainly examine the quantity, ratio and correlativity of concepts and relations to evaluate ontologies from the viewpoint of complexity and evolution, as well as analysing the concepts and their hierarchy in a conceptual model. The quantity of the ontology is assessed by determining the total number of concepts and relations in an ontology. The ratio metrics are used to determine the average number of relations per concepts, whereas, the correlativity consists of finding out how concepts are interrelated between them.

Another suite of metrics is proposed in Zhang, Li & Tan (2010) at both the ontology-level and class-level, to measure the design complexity of ontologies. These metrics are inspired by the concept of software metrics, and were analytically evaluated against Weyuker's criteria which are a set of properties for evaluating the usefulness of software complexity metrics (Weyuker 1988); these properties allow evaluating the behaviour of a metric. At the ontology level, the proposed measures include the size of the vocabulary of the ontology, the average number of relations per concept, the tree impurity and entropy of ontology graph. At the class level, these metrics are the number of children, class degree and depth of inheritance. The number of children of a given concept corresponds to the total number of concepts that directly inherit properties and attributes from this concept, whereas, the class degree of a concept is the total number of edges pointing to and leaving from this concept in the ontology graph.

2.4.2 Mathematical Analysis of Ontology Complexity

An ontology can be formally represented using a certain number of sets (Lozano-Tello et al. 2004; Kang et al. 2004; Tartir et al. 2005; Gangemi et al. 2005; Yang et al. 2006; Zhang et al. 2010; Durque-Ramos et al. 2013; etc…). These sets include: $C = \{c_1, c_2, c_3, \ldots c_m\}$ the set of m concepts of the ontology, $P = \{p_1, p_1, p_1, \ldots\ldots, p_n\}$ the set of n properties of the ontology, $R \subset P$ with $R = \{r_1, r_1, r_1, \ldots r_k\}$ the set of k relations of the ontology, $Att \subset P$ with $Att = \{att_1, att_1, \ldots\ldots att_p\}$ the set of p attributes of the ontology with $R \cup Att = P$, and $I = \{i_1, i_1, \ldots, i_r\}$ the set of r instances of concepts in the ontology.

2.4.2.1 Primitive Complexity Metrics

The metrics in this category are primitive because they are directly derived from the above sets representing the ontology. They mainly correspond to the number of elements of each set. These metrics are:

- |C| - total number of concepts in the ontology,
- |P| - total number of properties in the ontology,
- |R| - total number of relations in the ontology,
- |Att| - total number of attributes of concepts in the ontology and
- | I | - total number of instances in the ontology.

Additional to these metrics, Zhang, Li & Tan (2010) proposed a metric called Size of Vocabulary (SOV) that computes the amount of vocabulary defined in an ontology. This metric is given by the formula in Equation 2.1.

$$SOV = |C| + |P| \tag{2.1}$$

SOV measures the complexity of an ontology by counting its total number of entities. A greater SOV implies a greater size of ontology, and the time and effort required to build and maintain the ontology.

2.4.2.2 Schema and Inheritance Hierarchy Complexity Metrics

Yang et al. (2006), Yao (2006) and Zhang et al. (2010) proposed a group of metrics to measure the complexity of the inheritance hierarchy of an ontology, i.e., a sub-graph of an ontology graph composed of concepts and subclass relations. These metrics enables the determination of the following characteristics of an ontology: the longest and average path length, the average number of paths per concept, the longest path length of a concept and the average path length of a concept.

A path is defined by Yang et al. (2006) and Zhang et al. (2010) as a distinct trace that can be taken from a specific concept to a most general concept in the ontology, i.e., a concept without any parent or superclass. A path length is the sum of relations on that path. Another definition of a path is proposed by Yao et al. (2005) as a trace from a general concept to a leaf concept, i.e., a concept without any child. In simple terms, a path length corresponds to the sum of concepts on the path. These ontology characteristics are computed as follows.

- **Average number of paths per concept**

The value of this metric is obtained by dividing the sum of the number of paths of each concept by the total number of concepts in the ontology (|C|); it is given by the formula in Equation 2.2:

$$\rho = \frac{\sum_{i=1}^{m} p_i}{|C|} \tag{2.2}$$

where, p_i is the number of paths of a concept $C_i \in C$. This metric indicates the average connectivity degree of a concept (Yang et al. 2006) and it represents the level of usage of a concept by other concepts in the ontology. Therefore, a change in a concept may affect other concepts and vice versa. The value $\boldsymbol{\rho}$ for any ontology must be greater or equal to 1; a $\boldsymbol{\rho} = 1$ indicates that an ontology inheritance hierarchy is a tree. Zhang et al. (2010) proposed a similar metric called the Tree Impurity (TIP). This metric is used to measure how far an ontology inheritance hierarchy deviates from a tree. This is given by the formula in Equation 2.3:

$$TIP = |R'| - |C'| + 1 \tag{2.3}$$

where, R' is the set of relations in the inheritance hierarchy or the set of subclass relations of the entire ontology; and C' the set of concepts in the inheritance hierarchy.

According to Zhang et al. (2010), the rationale of the TIP metric resides in the fact that a well-structured ontology is composed of classes organized through inheritance relationships. TIP=0 means that the inheritance hierarchy is a tree. The greater the TIP, the more the ontology inheritance hierarchy deviates from the tree structure and the greater is its complexity.

- **Longest Path Length of Concept and Depth of inheritance**

The longest path length of a concept (C_i) and the depth of inheritance (Λ) are defined in Equations 2.4 and 2.5, respectively.

$$\lambda_i = \max(pl_{i,k}), 1 \le k \le p_i \tag{2.4}$$

$$\Lambda = \max(\lambda_i), 1 \le i \le m \tag{2.5}$$

where, $pl_{i,k} \in Pl_i = \{pl_{i,1}, pl_{i,2}, pl_{i,3}, \ldots pl_{i,pi}\}$ is the set of path length of a concept C_i. According to Zhang et al. (2010), a greater $\boldsymbol{\lambda_i}$ value shows that the class resides deeper in the inheritance hierarchy and reuses more information from its ancestors. A greater $\boldsymbol{\lambda_i}$ value also indicates that the class is more difficult to maintain as it is likely to be affected by changes in any of its ancestors. An ontology with a higher Λ is considered to have good semantic coverage of elements of the domain represented (Yang et al. 2006).

- **Average Path Length of a Concept C_i**

$$\bar{\lambda} = \frac{\sum_{k=1}^{p_i} pl_{i,k}}{p_i} \tag{2.6}$$

This metric defines the average number of ancestors of a given concept in the ontology. Thus, given a concept C_i, its average path length is obtained by dividing the sum of its path lengths (represented by $\sum pl_{i,k}$) by its number of paths (represented by p_i).

- **Average Path length of the Ontology**

$$\overline{\Lambda} = \frac{\sum_{i=1}^{m} \sum_{k=1}^{p_i} pl_{i,k}}{\sum_{i=1}^{m} p_i} \tag{2.7}$$

This metric defines the average number of ancestors of concepts in an ontology. An ontology with a higher average path length indicates the intensity of inheritance relationships amount its concepts. Therefore, the management and manipulation of concepts could be more complex in an ontology with higher average path length (Yang et al. 2006).

- **Entropy of Ontology Graph**

Kang et al. (2004) and Zhang et al. (2010) proposes a metric based on the Shannon's entropy function for determining the complexity of an ontology. In fact, the application of an entropy function to a probability distribution associated with graph elements (nodes and edges) provides a numerical value that can be used as an indicator of the graph complexity (Mowshowitz & Dehmer 2012). Given a graph, its entropy (EOG) is calculated as in Equation 2.8:

$$EOG = -\sum_{i=1}^{n} p(i) \log_2 p(i) \tag{2.8}$$

where, $p(i)$ is the probability for a concept to have i relations. This metric is used to assess the distribution of relations within the ontology; its maximum value corresponds to EOG = $log_2\ n$ when $p(i)=1/n$, and its minimum value EOG=0 is obtained when concepts have the same distribution of relations i.e. all nodes of the ontology sub-graphs have the same number of edges. Therefore, an ontology with a lower EOG can be considered as less complex in terms of relations distribution.

2.4.2.3 Concept and Relations Complexity Metrics

Another group of metrics used to analyse ontology complexity is the one helping assess how concepts are interrelated in an ontology. These metrics can be used to determine the average number of properties each concept has in an ontology, the average number of attributes per concept, the average number of relations, and the sub-class and part-of relations of a concept in an ontology.

- **Average Number of Relations and Attributes per Concept**

The average number of relations per concept (ANR) is one of the most used metric for ontology evaluation. It provides an indication of the ontology complexity since a concept is related to other concepts (Supekar et al. 2004, Yang et al 2006 and Zhang et al. 2010). The ANR is computed with the formula in Equation 2.9.

$$ANR = \frac{|R|}{|C|} \tag{2.9}$$

where, R represents the set of relations and C the set of concepts of the ontology. Kang et al. (2004), Tartir et al. (2005) and Duque-Ramos et al. (2013) proposed a metric for determining the average number of attributes per concept (ANA); this is obtained by dividing the total number of attributes by the total number of concepts in the ontology as in Equation 2.10:

$$ANA = \frac{|Att|}{|C|} \tag{2.10}$$

where C is the set of concepts and Att the set of attributes of concepts in C. An ontology with a higher ANA indicates the degree of richness of information per concept, whereas, a lower ANA value might indicate the low information availability for each ontology concept. According to Kang et al. (2004), the number of attributes per concept is among the main factors influencing the complexity of an ontology.

- **Average Number of Subclasses per Concept**

Tartir et al. (2005) proposed a metric for determining the average number of sub class relations per concept. This metric called inheritance richness belongs to the OntoQA (Tartir et al. 2005) ontology evaluation method and tool described early in this chapter. The inheritance richness of a concept is given in the formula in Equation 2.11:

$$IR_S = \frac{\sum_{c_i \in C} | H^C(C_j, C_i) |}{| C |} \qquad (2.11)$$

where, H^C (C_j , C_i) is a function representing a taxonomy relation between the classes C_j and C_i; therefore $|H^C$ (C_j , C_i)$|$ is the number of sub-classes of a class $C_{i.}$. The number IR_s describes the distribution of information across different levels of the ontology's inheritance tree.

- **Average Number of Super Classes per Concept**

Duque-Ramos et al. (2013) proposed a metric called Tangledness of ontology (TMOnto). This metric represents the average number of parents per concept within the inheritance hierarchy. According to Lu (2006), the number of parents of a concept indicates how many parents a child concept inherits from. The TMOnto is computed as in Equation 2.12:

$$TMOnto = \frac{| R |}{| C | - | C(DP) |} \qquad (2.12)$$

where, C (DP) is the set of classes in the ontology with more than one direct parent. The TMOnto can be considered as the inverse of the IR_s (Equation (11)); therefore, another formula for the average number of parents (ANP) per concept can be deduced using the IR_s formula (Equation 2.11) as in Equation 2.12.

$$ANP = \frac{\sum_{c_i \in C} | H^C(C_j, C_i)^{-1} |}{| C |} \qquad (2.13)$$

where, H^C (C_j , C_i)$^{-1}$ is the inverse of the function H^C (C_j , C_i); therefore $|H^C$ (C_j , C_i)$^{-1}|$ is the number of super-classes of the class $C_{j.}$ The value of ANP represents the average number of concepts from which a given concept inherits some of its characteristics (Attributes, properties, etc.). An ontology with a higher ANP indicates a high degree of interrelations between the concepts in this ontology. Therefore, a change to any concept may affect other concepts in the ontology (Lu 2006).

- **Percentage of Part-of Relations in the Total Number of Relations or relationship richness**

The part-of relations which are the aggregation or composition of relations in an ontology also provides an indication of the ontology complexity. An ontology with a high number of inheritance relations is more complex to reuse and maintain. An ontology with a high percentage of part-of relations can be considered less complex. In Tratir et al. (2005), the relationship richness (RR) has been proposed as a metric for determining the percentage of part-of relations; its definition is provided in Equation 2.14:

$$RR = \frac{|P|}{|SC| + |P|} \tag{2.14}$$

where, | P | and |SC| are the total number of properties and the total number of SubClassOf relations, respectively. The RR metric provides an indication on the diversity and placement of relations in the ontology.

2.5 CONCLUSION

This chapter conducted a review of the literature on ontologies. Ontology and its components including concepts or classes, properties, axioms, restrictions, rules and inheritance hierarchy were defined. Thereafter, the different approaches for evaluating ontologies as well as existing frameworks for ontology evaluation provided in the literature were reviewed. Finally, the existing metrics for analysing the complexity of ontologies were discussed in detail. The next chapter presents the material and methods used in this study.

CHAPTER 3: MATERIAL AND METHODS

3.1 INTRODUCTION

The activity of analysing the ontologies of a domain to guide their selection and reuse includes three main steps: online acquisition of ontologies, ontology assessment and ontology comparison (Suarez-Figueroa, Gomez-Perez & Fernandez-Lopez 2010). This chapter presents the techniques employed to gather the biomedical ontologies that constitute the dataset in this study. Thereafter, the design and implementation of the algorithms for computing the complexity metrics of these ontologies through the processing of their RDF graphs are presented. Finally, the decision making methods for comparing and ranking of the ontologies are presented.

3.2 ONLINE ACQUISITION OF ONTOLOGIES

In this study, biomedical ontologies were collected from the Bioportal repository. The BioPortal repository includes 491 biomedical ontologies and provides tools and services for browsing the ontologies. Developed during the early 2000, BioPortal is a community-based ontology repository for biomedical ontologies where users can publish, submit new versions, browse, and access the ontologies and their components through a set of REST services and SPARQL (Salvadores, Alexander, Musen and Noy 2013). The Web interface of BioPortal allows users to browse the list of ontologies, search and comment on the terms in the ontologies, annotate text with ontology terms, and search an ontology-based index of biomedical resources (Whetzel and Team 2013). Ontologies in the BioPortal are grouped into 18 categories such as: anatomy, chemical, health, human, immunology, molecule, protein, taxonomic classification, and so on (Wbio 2015). However, if a new ontology falls in a category that does not exist, the administrator of the ontology can register a new category (Salvadores et al. 2013). The algorithms for computing the complexity metrics of the collected ontologies are presented in the next section.

3.3 ALGORITHMS FOR COMPUTING ONTOLOGY COMPLEXITY METRICS

This section presents the algorithms designed for the computation of ontology complexity metrics. These algorithms are grouped into 3 categories based on the metrics they compute; these include: path-related, entropy and class and relation richness algorithms. Prior to the presentation of these algorithms, some definitions are provided along with a presentation of the Jena API toolkit used to implement these algorithms.

3.3.1 Jena API Implementation of RDF Ontology

Jena API (Application Programming Interface) is an open source Library for developing semantic web applications through extraction and manipulation of RDF graphs of ontologies. Jena API Library includes interfaces for RDF and OWL ontologies, a SPARQL engine and RDF parsers. In Jena, a RDF graph is represented by the Model interface which represents the set of statements of RDF ontology. The Model interface also provides functions for retrieving and saving RDF graphs from and to files as well as functions for creating resources, properties and literals, and the statements for linking them. Other Jena interfaces include the OntClass interface representing a node of a RDF graph, the Resource interface which represents a URI (Unified Resource Identifier), the Property interface for the ontology properties, etc. The Jena Library also provides interfaces to access various database management systems such as Oracle, MySQL, PostgreSQL (Wlkinson et al. 2003).

3.3.2 Entities for Ontology Representation

This section defines the underlying concepts used for the processing of RDF graphs in this study. An RDF document or graph is a collection of triples (subject, predicate and object) that can be seen as a direct multigraph, that is, two nodes can be connected by more than one edge; where classes and properties are the nodes and edges, respectively. An RDF graph G is a tuple <C, P> where C and P are the sets of classes/nodes and properties/edges, respectively. The inheritance hierarchy of the RDF graph G is a subgraph G'. G' is also a tuple <C', P'> where C' is the set of classes and P' the set of properties in G'.

A path t between two nodes c_0 and c_n in G' is represented as in Equation 3.1 and is defined as a sequence of unrepeated nodes connected by edges (properties) from c_0 to c_n; the length pl_t of this path is the number of edges on the path.

$$t = c_0 \rightarrow c_1 \rightarrow c_2 \rightarrow \cdots \rightarrow c_n \qquad (3.1)$$

A path between a root node and a node c_i ($0 \leq i \leq n$) is called path of c_i. The total number of paths (p_i) between root nodes and other nodes c_i is determined using the function p on C' as in Equation 3.2.

$$p: C' \rightarrow \mathbb{N},\ p(c_i) = p_i \qquad (3.2)$$

The set of subclasses of a node c_i is determined through the function h in Equation 3.3.

$$h: C' \rightarrow C',\ h(c_i)=\{c \in C',\ c \text{ is subclass of } c_i\} \quad (3.3)$$

The set of superclass of a node c_i is obtained with h^{-1}, the inverse of *h as* in Equation 3.4.

$$h^{-1}: C' \rightarrow C',\ \ h^{-1}(c_i)=\{c \in C',\ c \text{ is superclass of } c_i\} \quad (3.4)$$

The degree E (c_i) of a node c_i is the sum of its number of superclasses and subclasses in G' and is given in Equation 3.5.

$$E\ (c_i) = card(h(c_i)) + card(h^{-1}(c_i)) \quad (3.5)$$

3.3.3 Proposed Algorithms

This section presents the proposed algorithms for computing ontology complexity metrics. There are in total nine algorithms organised into three main groups, namely, path-related, entropy, and class and relation richness algorithms as presented in the following subsections.

3.3.3.1 Path-related Algorithms

This subsection presents four algorithms developed for the computation of the average number of paths per class, the average path length and the tree impurity. To compute the average number of paths per class and the average path length, Algorithm 1 that uses Algorithms 2, 3 and 4 is used.

Algorithm 1 (*FINDNUMBERPATHS*) processes the ontology Model and the depth of inheritance (obtained from Bioportal together with the ontology) to obtain a set of paths of leaf nodes in the RDF graph of ontology (*FINDNUMBPATHS* from line 4 to line 16). The resulting set of paths is used to get the average number of paths per class and the average path length (lines 17-21 and 32-33 of *FINDNUMBPATHS*). The tree impurity is obtained through the counting of the root nodes, subclass of relations and nodes belonging to the inheritance hierarchy (*FINDNUMBPATH* lines 8-9, 23-25, 28-29 and 34).

Algorithm 1: FINDNUMBPATHS

```
1.    Input: Jena Ontology Model (M), depth,
2.    Output: averageNumbPath, averagePathLength, treeImpurity
3.    Begin
4.      Create setOfPaths
5.      Create listOfPaths
6.      countSubClassOfRelation ←0, classtree ←0, rootclass ←0
7.      For each class c_i of M Do
8.        If card(h^-1(c_i))=0 Then
9.          rootclass ←rootclass+1
10.         Create pathNode
11.         pathNode.add (c_i)
12.         setOfPaths.add (pathNode)
13.       EndIf
```

```
14.    EndFor
15.   listOfPaths ← TOTALPATHS (setOfPaths, M, depth)
16.   listOfPaths ← DUPLICATE (listOfPaths)
17.   For each class c_i of M Do
18.     pathResult ← PATH (listOfPaths, c_i , depth)
19.   averageLengthC_i ← pathResult[1]/pathResult[0]
20.   pathLength[0]=pathLength[0]+pathResult[0]
21.   pathLength[1]=pathLength[1]+pathResult[1]
22.    EndFor
23.     For each ontology statement S in M Do
24.       If S ∈ R' Then
25.         countSubClassOfRelation ← countSubClassOfRelation + 1
26.     EndFor
27.    For each class c_i of M Do
28.     If c_i ∈ C' Then
29.       classtree ← classtree+1
30.     EndIf
31.   EndFor
32.    averageNumbPath=pathLength / |C|
33.     averagePathLength = pathLength[0]/pathLength[1]
34.     treeImpurity = (countSubClassOfRelation – classtree) + rootclass + 1
35.   End
```

Formally, Algorithm 1 works as follows: A set of paths (SetOfPaths) is created (*FINDNUMBPATHS* from line 4), and each subset of SetOfPaths is initialized with a root node (*FINDNUMBPATHS* from line 4 to line 12). SetOfPaths is then used along with the Jena Model of the ontology and the value of the depth of inheritance (line 15 of *FINDNUMBPATHS*) as parameters to Algorithm 2 (*TOTALPATHS*) which returns another set of paths ListOfPaths. ListOfPaths is further passed as a parameter to Algorithm 3 (*DUPLICATE*) to remove the duplicated sets of nodes from the list of paths (line 16 of *FINDNUMBPATHS*). The ontology classes in ListOfPaths returned by *DUPLICATE* are fed together with the value of the depth of inheritance to Algorithm 4 (*PATH*) which returns an array containing the number of paths of input class and the sum of lengths of its paths.

The outputs of *PATH* are then used to determine the average length of paths of the class (*FINDPATHS* lines 19), the sum of the number of paths of all the classes (*FINDPATHS* lines 20) and the sum of the lengths of all the paths (*FINDPATHS* lines 21).

Algorithm 2: TOTALPATHS

```
Input: Jena Model (M), setOfPaths, depth
Output: L
Begin
 While ¬Empty (setOfPaths)
   setOfNodes ← remove last element of setOfPaths
   listOfPaths.add(setOfNodes)
   If Size of setOfNodes < depth Then
     Node= last element (setOfNodes)
     I= h(Node)
   For each s ∈ I Do
       setOfPaths.add(setOfNodes.add(subject(s))
   EndFor
 EndIf
EndWhile
Return (listOfPaths)
End
```

Algorithm 3: DUPLICATE

```
Input: listOfPaths
Output: listOfPaths
Begin
 For i←0 to i<= size of listOfPaths – 1 Do
   j← i+1
   While j<= size of listOfPaths or f=false
     If listOfPaths (i) is contained into listOfPaths (j)
       Delete listOfPaths (i) from listOfPaths
       f←true
     EndIf
     j← j+1
   EndWhile
 EndFor
 Return (listOfPaths)
End
```

If the size (number of nodes) of SetOfNodes is less than the depth of inheritance (line 7) a group of instructions from line 8 to 13 are executed to find the subclasses of the last element of SetOfNodes (line 8 and line 9). Each subclass of the last element is added to SetOfNodes, which in turn is added to SetOfPaths in line 11. Algorithm 3 (*DUPLICATE*) removes from the list of paths ListOfPaths returned by *TOTALPATHS* the set of duplicated nodes. *DUPLICATE* uses an iterative process from line 4 to line 13. Two counters are used at each iteration to test and remove duplicated nodes (line 4 to 11).

Algorithm 4: PATH

```
1.  Input: listOfPaths, a class ci, depth
2.  Begin
3.   countNumbPaths← 0, countPathsLength←0
4.    Create pathsci
5.    i← 0
6.    For each p ϵ listOfPaths Do
7.      If p contains ci Then
8.        pathsci.add(p)
9.      EndIf
10.   EndFor
11.   While i<= depth
12.      Create posPathsci
13.      For each p ϵ pathsci Do
14.       If p.(i)= ci Then
15.          posPathsci.add(p)
16.      EndIf
17.    EndFor
18.    For j←0 to j<= card(posPathsci) - 1
19.       For k←j+1 to k<= card(posPathsci)
20.         If ((SubSet (posPathsci(j),i))=(SubSet (posPathsci(k),i)))
21.            Delete posPathsci(k) from posPathsci
22.         EndIf
23.      EndFor
24.    EndFor
25.      countPathsLength = countPathsLength +(card(posPathsci)*i)
26.      countNumbPaths = countNumbPaths + card(posPathsci)
27.      i←i+1
28.    EndWhile
29.    pathLength[0]← countPathsLength
30.    pathLength[1]← countNumbPaths
31.    Return pathLength
32. End
```

Algorithm 4 (*PATH*) is executed with the list of paths without duplicates listOfPaths returned by the *DUPLICATE* and the Jena Model of the ontology. For every class $c_i \in C'$ a set of instructions is executed from line 6 to line 27 to determine the number of paths and sum of path lengths. In line 5 a set of paths $pathsc_i$ is created and filled with elements of *listOfPaths* containing the current class c_i (line 7). In line 11 a loop is executed until the value of a counter is equal to the depth of the ontology. Within the loop another set of paths $posPathsc_i$ is created (line 12) and filled with elements of $pathsc_i$ where there is a match with classes at the position of the loop counter (line 15). Thereafter, iterations are executed from line 18 to line 24 to remove the duplicated paths from posPathsc$_i$.

3.3.3.2 Entropy Algorithms

This set of algorithms include Algorithms 5 to 8. Algorithm 5 (*ENTROPY*) calls the Algorithm 6 (*NUMBEDGES*) which in turn calls Algorithms 7 (*MAXNUMBEDGES)* and 8 (*TOTALEDGES)*. The *ENTROPY* receives as input the number of classes of the ontology and the Jena Model of the ontology; it uses the list returned by *NUMBEDGES* (Line 4 *ENTROPY*) to compute the entropy of the ontology graph (lines 5 to 12).

Algorithm 5: ENTROPY

```
1.   Input: number of classes (n), Jena Ontology Model (M)
2.   Begin
3.     Create a List of Integer (F)
4.     F ← NUMBEDGES (M)
5.     For j ←0 to j<= size of F Do
6.       If (F.get(j)>0)
7.         P ← F.get(j)/n //Probability for a class to have j edges
8.         V ← Log2 P
9.         entropy ←entropy + P*V
10.      EndIf
11.    EndFor
12.    Return (entropy *(-1))
13.  End
```

The *ENTROPY* starts by creating a list of integers in line 3; this list is populated by *NUMBEDGES* (*ENTROPY* line 4). An iterative process is executed (*ENTROPY* lines 5 to 11) to test the value of each edge of the ontology graph (*ENTROPY* line 4); this value is then divided by the number of classes to obtain the probability for a class c_i to have *i* relations in the ontology (*ENTROPY* line 9). In lines 10 and 11 the calculation of the entropy of the ontology graph is completed and its value is multiplied by -1 and returned in line 12.

Algorithm 6: NUMBEDGES

```
1.   Input: Jena Ontology Model (M)
2.   Begin
3.     max ← MAXNUMBEDGES (M)
4.     Create a List of Integer (F) with size max+1
5.     For i ←0 to i<= max Do
6.       For each class ci in M Do
7.           totalNumberEdges ← TOTALEDGES (M, ci)
8.           F.set(totalNumberEdges, F.get(totalNumberEdges) + 1)
9.       EndFor
10.    EndFor
11.    Return F
12   End
```

Algorithm 7: MAXNUMBEDGES

```
1.   Input: Jena Ontology Model (M)
2.   Begin
3.     maxEdges ←0
4.     For each class ci in M Do
5.       totalNumberEdges ← TOTALEDGES (M, ci)
6.       If (maxEdges < totalNumberEdges) Then
7.         maxEdges ← totalNumberEdges
8.       End If
9.     EndFor
10.    Return maxEdges
11   End
```

In Algorithm 6 (*NUMBEDGES)* the total number of edges (max) in the ontology graph is obtained with Algorithm 7 (*MAXNUMBEDGES)* in line 3. This number is then used to create a list of integers with the size equal to the number plus one (line 4). Thereafter, an iterative process is applied (line 5 to 9) to determine the degree of each class c_i, $E(c_i)$ with Algorithm 8 (*TOTALEDGES)*. *MAXNUMBEDGES* determines the maximum degree value in the ontology graph. An iterative process from line 4 to 9 determines the degree $E(c_i)$ of each class c_i in the ontology graph using *TOTALEDGES* (*MAXNUMBEDGES* lines 4 and 5); the values obtained are iteratively compared amongst themselves to determine the bigger one (*MAXNUMBEDGES* lines 6 and 7).

Algorithm 8: TOTALEDGES

```
1.  Input:  Jena Ontology Model (M), ontology class (c_i)
2.  Begin
3.   noSuperClasses ← 0
4.   noSubClasses ← 0
5.   K ← M.listStatements() //statements in M where c_i is an Object
6.   For each statement in K Do
7.     noSubClasses += 1
8.   EndFor
9.   K ← M.listStatements() //statements in M where c_i is a Subject
10.  For each statement in K Do
11.    noSuperClasses += 1
12.  EndFor
13.  Return (noSuperClasses + noSubClasses)
14. End
```

The *TOTALEDGES* is executed with two parameters the Jena Model of the ontology and a class c_i of this ontology; it determines and returns the degree $E(c_i)$ of the class c_i.

3.3.3.3 Class and relation richness algorithm

Algorithm 9 (*RICHNESS)* counts the number of instances of classes in the ontology graph (lines 4 to 7); this number is further divided by the total number of classes in the ontology to obtain the value of the class richness (*RICHNESS* line 8). The computation of the relation richness (RR) starts in line 9 of the *RICHNESS* by collecting all the statements of the ontology. These statements are then tested from lines 10 to 16. The test determines the total number of subclassOf relations (*RICHNESS* line 11) and the number of relations other than subclassOf which are represented in the form of restrictions (*RICHNESS* line 14). The number of subclassOf and other relations are used to compute the RR in line 19 of *RICHNESS*.

Algorithm 9: RICHNESS

```
1.  Input: number of classes (nClasses), Jena Ontology Model (M)
2.  Output: class richness, relationship richness
3.  Begin
4.  For each class c_i of M Do
5.  If c_i has an instance Then
6.  countInstances ← countInstances+1
7.  EndIf
8.  classRichness ← countInstances/nClasses
9.  For each ontology statement S in M Do
10. If S is a SubclassOf relation Then
11. subClassOfRel ← subClassOfRel + 1
12. Else
13. If S is a Restriction Then
14. otherRel ← +1
15. EndIf
16. EndIf
17. EndFor
18. relations ← otherRel+ subClassOfRel
19. relationRichness ← relations/(relations+ subClassOfRel)
20. End
```

3.4 RANKING OF ONTOLOGIES

To rank the ontologies in the dataset based on their complexity metrics, 5 Multi-criteria Decision Making Process (MCDM) methods are used in this study including: Weighted Sum Model (WSM), Weighted Product Model (WPM), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), Weighted Linear Combination Ranking Technique (WLCRT) and ELECTRE (Benayoun et al. 1966; Fishburn 1967; Miller et al. 1969; Saaty 1977; Chou 2013). These 5 MCDMs methods are explained in detail in the next subsections.

3.4.1 Multi-criteria Decision Making Process

According to Zimmerman (1991), MCDM processes can be divided into two groups: multi-objectives decision making (MODM) and multi-attributes decision making (MADM). The first group applies to decision problems in which the decision space is continuous. A typical example is mathematical programming problems with multiple objective functions. The second group concentrates on problems with discrete decision spaces. In these problems the set of decision alternatives is predetermined. This research falls in the second group as the application of MADM consists of ranking ontologies based on their predetermined complexity metrics.

Independent of their numbers, all MADM have a number of characteristics in common such as the alternatives, attributes or criteria, criteria weights and decision matrix (Chen and Hwang 1992). These common features of MADM are defined below.

- ***Alternatives*** – They are the different choices of action available to the decision maker. Usually, the set of alternatives is assumed to be finite, ranging from several to hundreds. They are supposed to be screened, prioritized and eventually ranked. In a decision making problem the set of M alternatives is given by $A= \{A_1, A_2, A_3, \dots, A_{M-1}, A_M\}$
- ***Attributes or Criteria*** **-** Represent the different dimensions from which the alternatives can be viewed. In a decision making problem the set of N criteria is given by $C=\{C_1, C_2, C_3, \dots, C_{N-1}, C_N\}$
- ***Criteria weights*** **-** Most of the MADM involve the determination and use of the weight or importance level of each of the criteria. They help determine how an attribute is more or less important than another. Usually the sum of the criteria weights is equal to 1. The set of criteria weights is given by the vector $W=(w_1, w_2, w_3, \dots, w_{N-1}, w_N)$, with

$$\sum_{j} wj = 1 \tag{3.6}$$

For the sake of consistency, the same set of criteria weights is used for each of the 5 MADM in this study. These weights are determined with the WLCRT method.

- ***Decision matrix*** - A MADM problem with M alternatives and N criteria is usually represented in the form of a matrix called decision matrix. A decision matrix D is a matrix of $M \times N$ dimensions where each element d_{ij} corresponds to the performance of the alternative A_i when it is evaluated in terms of decision criterion Cj, (for i = 1,2,3,..., M, and j = 1,2,3,..., N). A decision matrix is drawn in Equation 3.7.

$$D = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ \vdots \\ \vdots \\ A_m \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & \cdots & C_n \end{array} \\ \begin{bmatrix} d_{11} & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & d_{22} & d_{23} & \cdots & d_{2n} \\ d_{31} & d_{32} & d_{33} & \ddots & d_{3n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & d_{m3} & \cdots & d_{mn} \end{bmatrix} \end{array} \tag{3.7}$$

In this study, the alternatives are the ontologies in the dataset, whereas, the attributes or criteria are the complexity metrics of these ontologies.

3.4.2 Weighted Sum Model

The Weighted Sum Model (WSM) is considered to be most used and easiest approach for implementing a decision making process (Triantaphyllou 1998). Formally, it consists in assigning to an alternative A_i a score that corresponds to the sum of the products of each of its performances and their respective weights. A score of an alternative A_i is obtained as in Equation 3.8.

$$Score(A_i) = \sum_{j}^{N} d_{ij} \cdot w_j \quad \text{(for } i = 1,2,3... M\text{, and } j = 1,2,3,..., N\text{)} \tag{3.8}$$

where, *dij* is the element of the decision matrix *D* and it corresponds to performance of the *Ai* alternative for the *Ci* criterion (Equation 3.8). In case where the decision matrix is normalised into a normalised decision matrix *R (M×N)* with elements r_{ij}, Equation 3.8 is transformed into Equation 3.9.

$$Score(A_i) = \sum_{j}^{N} r_{ij} \cdot w_j \quad \text{(for } i = 1,2,3... M\text{, and } j = 1,2,3,..., N\text{)} \qquad (3.9)$$

3.4.3 Weighted Product Model

The Weighted Product Model (WPM) is similar to the WSM. The difference between these two MADM lies in the fact that instead of adding up the products of performances to the criteria weights, the score of an alternative A_i is obtained by multiplying the exponential of each of its performances to their respective weights. A score of an alternative A_i in the WPM method is given in Equation 3.10.

$$Score(A_i) = \prod_{j}^{N} d_{ij}^{\ w_j} \quad \text{(for } i = 1,2,3... M\text{, and } j = 1,2,3,..., N\text{)} \qquad (3.10)$$

Similar to the WSM, in case where the decision matrix is normalised into a normalised decision matrix *R (M×N)* with elements r_{ij}, Equation 3.10 is transformed into Equation 3.11.

$$Score(A_i) = \prod_{j}^{N} r_{ij}^{\ w_j} \quad \text{(for } i = 1,2,3... M\text{, and } j = 1,2,3,..., N\text{)} \qquad (3.11)$$

3.4.4 Technique for Order Preference by Similarity to Ideal Solution

The Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) is an algorithm for finding the best solution among all practical alternatives; it takes into consideration both the shortest distance from the Positive Ideal Solution (PIS) and the farthest distance from the Negative Ideal Solution (NIS), and preference order is ranked according to their relative closeness (Hwang et al. 1981). The TOPSIS method has been widely implemented for different decision making problems in various areas such as: risk assessment, customer evaluation, weapon selection, and performance evaluation (Jiang et al 2010).

The TOPSIS is an algorithm that consists of 6 steps:

- Calculation of the normalized decision matrix,
- Calculation of the weighted normalized matrix,
- Determination of the PIS and NIS,
- Calculation of the separation measures,
- Calculation of the relative closeness to the ideal solution and
- Ranking of alternatives.

The abovementioned steps that constitute the TOPSIS algorithm are explained in detail in the next subsections.

3.4.4.1 Calculation of the Normalized Decision Matrix

The normalized decision matrix *R* in TOPSIS is obtained by replacing every element of the initial decision matrix *D* (Equation 3.7) by its ratio to the square root of the sum of the squares of all elements situated in the same column with the element to be replaced. This calculation of is formally represented in Equation 3.12.

$$r_{ij} = \frac{d_{ij}}{\sqrt{\sum_{i}^{M} d_{ij}^{2}}} \quad \text{(for } i = 1,2,3...\ M\text{, and } j = 1,2,3,...,\ N\text{)} \qquad (3.12)$$

3.4.4.2 Calculation of the Weighted Normalized Matrix

The weighted normalized matrix *V* is obtained through with Equation 3.13.

$$\forall\ v_{ij} \in V,\ v_{ij} = r_{ij} \cdot w_j \quad \text{(for } i = 1,2,3...\ M\text{, and } j = 1,2,3,...,\ N\text{)} \qquad (3.13)$$

The matrix *V* is obtained by multiplying the elements of each column of the normalized decision matrix *R* by the weight of their corresponding criterion.

3.4.4.3 Determination of the PIS and NIS

This consists of determining two different sets from the weighted normalised matrix *V*. The first set (A^+) which is the PIS includes the maximum values in each column of *V* and the second set (A^-) representing the NIS contains the minimum values in each column of *V*. A^+ and A^- are formally expressed in Equations 3.14 and 3.15, respectively.

$$A^+ = \{ v_1^+, v_2^+, \ldots, v_n^+ \} = \{\max v_{ij}, j = 1,.....,n\} \quad (3.14)$$

$$A^- = \{ v_1^-, v_2^-, \ldots, v_n^- \} = \{\min v_{ij}, j = 1,.....,n\} \quad (3.15)$$

3.4.4.4 Calculation of the Separation Measures

The separation measures of every alternative are calculated here. For each alternative, both its distances from the PIS (d^+) and NIS (d^-) are computed using the weighted normalised decision matrix (Equation 3.13) and the sets of positive and negative ideal solutions (Equation 3.14 and 3.15). The distances (d^+) and (d^-) are computed with Equation 3.16 and 3.17, respectively.

$$d_i^+ = \sqrt{\sum_{j=1}^{m}(v_{ij} - v_j^+)^2} \quad \text{(for } i = 1,2,3... M, \text{ and } j = 1,2,3,..., N) \quad (3.16)$$

$$d_i^- = \sqrt{\sum_{j=1}^{m}(v_{ij} - v_j^-)^2} \quad \text{(for } i = 1,2,3... M, \text{ and } j = 1,2,3,..., N) \quad (3.17)$$

where, $v_{ij} \in V$, $v_j^+ \in A^+$ and $v_j^- \in A^-$.

3.4.4.5 Calculation of the Relative Closeness to the Ideal Solution

The relative closeness value c_i of an alternative A_i is computed from its distances to the positive and negative ideal solutions (Equations 3.16 and 3.17). It is given by:

$$c_i = \frac{d_i^-}{d_i^+ + d_i^-} \quad \text{(for } i = 1,2,3... M) \quad (3.18)$$

3.4.4.6 Ranking of Alternatives

The alternatives are ranked based on their relative closeness values (Equation 3.18).

3.4.5 Weighted Linear Combination Ranking Technique

The Weighted Linear Combination Ranking Technique (WLCRT) proposed by Chou (2013) is based on the linear combination of matrix algebra calculations. Unlike the WSM, WPM and TOPSIS, the WLCRT algorithm defines a specific procedure for determining the criteria weights based on the calculation of the Pearson correlation coefficient and the eigenvector method. The other peculiarity of the WLCRT method resides in the determination of the alternative scores.

The WLCRT algorithm consists of 5 steps:

- Construction of the normalized decision matrix,
- Elicitation of criteria weights,
- Aggregation of the preference or alternative information,
- Ranking of alternatives and
- Sensitivity analysis.

The abovementioned steps of the WLCRT algorithm are explained in detail in the next subsections.

3.4.5.1 Construction of the Normalized Decision Matrix

The decision matrix D' in WLCRT is obtained by transforming the decision matrix *D* in Equation 3.7 in two steps. The first step consists of computing the elements d_{ij}' of D' from the elements of *D* with Equation 3.19.

$$d_{ij}' = \frac{d_{ij} - d_{j\min}}{d_{j\max} - d_{j\min}} \quad \text{(for } i = 1,2,3... M\text{, and } j = 1,2,3,..., N\text{)} \quad (3.19)$$

A d_{ij}' equal to 0 will be transformed into 0.1 while the one equal to 1 will be set to 0.9. The second step consists of computing the normalized decision matric *R* of *D* based on D' as in Equation 3.20.

$$r_{ij} = 0.1 + 0.8d_{ij}' \text{ (for } i = 1,2,3... M, \text{ and } j = 1,2,3,..., N) \qquad (3.20)$$

3.4.5.2 Elicitation of Criteria Weights

The determination of the weights criteria begins with the calculation of the Pearson correlation coefficients from the normalised decision matrix *R*. The Pearson correlation coefficient *correl(x,y)* of two discrete variables x=[x_1, x_2,...., x_{n-1}, x_n] and y=[y_1, y_2,...., y_{n-1}, y_n] is a value that expresses the distance (or linear dependence) between these variables (Hauke and Kossowski 2011). It is used to determine whether 2 variables are related. The Pearson correlation coefficient is calculated with Equation 3.21.

$$correl(x,y) = \frac{\sum_i^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_i^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_i^n (y_i - \bar{y})}} \qquad (3.21)$$

where, $-1 \le correl(x,y) \le 1$; $\bar{x} = \frac{1}{n}\sum_i^n x_i$ and $\bar{y} = \frac{1}{n}\sum_i^n y_i$

Therefore, if each column of the normalised decision matrix *R* is a discrete variable, the Pearson correlation coefficients between M criteria of *R* form a proximity matrix C ($M \times M$) as in Equation 3.22.

$$C = \begin{bmatrix} 1 & c_{12} & c_{13} & \cdots & c_{1M} \\ c_{21} & 1 & c_{23} & \cdots & c_{2M} \\ \vdots & \vdots & 1 & \ddots & \vdots \\ \vdots & \vdots & \vdots & 1 & \vdots \\ c_{M1} & c_{M2} & c_{M3} & \cdots & 1 \end{bmatrix} \text{(for each } c_{ij} \in C, c_{ij}=1 \text{ if } i=j; c_{ij}= c_{ji} \text{ if } i \neq j) \; (3.22)$$

The proximity matrix C expresses a set of observations on how correlated the criteria are. According to Chou (2011), the weights of the criteria represent the priorities of the elements of the principal diagonal of the proximity matrix; these weights are the absolute values of the eigenvectors that correspond to the maximum eigenvalue λ_{max}. Given a linear transformation (or linear matrix) P, a non-zero vector *w* is defined as an eigenvector of C if there is a scalar λ that satisfies the Equation 3.23.

$$Pw = \lambda w \qquad (3.23)$$

where, the scalar λ is called the eigenvalue of C for the eigenvector *w* (Moghddam et al. 1994). If P is a square matrix of dimension 4x4, Equation 3.23 can be represented as in Equation 3.24.

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \lambda \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \tag{3.24}$$

Equation 3.24 can be transformed into Equation 3.25.

$$\begin{bmatrix} p_{11}-\lambda & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22}-\lambda & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33}-\lambda & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44}-\lambda \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.25}$$

Given the set of eigenvectors and their corresponding eigenvalues, one can, according to the Eigen decomposition (Arbenz 2016), obtain a diagonal matrix where each element of the diagonal corresponds to an eigenvalue. This matrix is given in Equation 3.26.

$$W^{-1} \cdot P \cdot W = diagonal\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\} \tag{3.26}$$

Where, W is a matrix composed of the eigenvectors of P and W^{-1} the inverse of W.

3.4.5.3 Aggregation of the Preference or Alternative Information

The aggregation of preference or alternatives consists in transforming a set of numerical values into a unique representative value of an alternative (Smolikova et al. 2002). An aggregation is a continuous function $h: [0, 1]^n \rightarrow [0, 1]$ that determines the unique value of an alternative (Smolikova et al. 2002). Given the weights of criteria of a decision making process, the aggregation operator *h* in the WLCRT method is defined as in Equation 3.27; it a parameter function called the weighted generalised means.

$$h_\alpha(A_i) = (\sum_j^N w_j \cdot r_{ij})^{\frac{1}{\alpha}} \text{ (for } i = 1,2,3\dots M\text{, and } j = 1,2,3,\dots, N) \qquad (3.27)$$

where, A_i is an alternative, w_j the weight of criterion C_j and r_{ij} the performance of the alternative A_i to a criterion C_j. α (-∞< α <+∞) is a non-zero real number, it is the parameter of the aggregation operator h. By varying α with a constant value Δα in the interval [-∞, +∞], one can obtain a curve of the function h_α as in Figure 3.1.

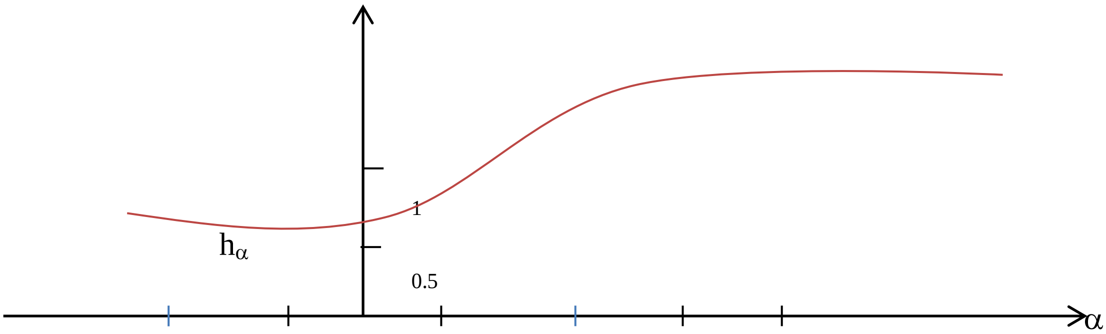

Figure 3.1 Sample Curve of h_α

The score of the alternative A_i corresponds to the mean or average of h_α (Monea 2015) and it is calculated as in Equation 3.28.

$$\bar{h}_\alpha = \frac{\int_a^b h_\alpha d\alpha}{(b-a)} \qquad (3.28)$$

where, $\bar{h}_\alpha$ is the mean value of h_α, a and b the beginning and end of an arbitrary interval [a, b] ⊆ [-∞, +∞]. By approximating the space between the curve of h_α and the α-axis in Figure 3.1 one obtains a trapezoid; therefore, the trapezoidal rule is used to approximate the value of $\int_a^b h_\alpha d\alpha$. The trapezoidal rule is defined as in Equation 3.29.

$$\int_a^b h_\alpha d\alpha \cong \Delta\alpha . \left[\frac{h_a + h_b}{2} + \sum_{i=1}^{u-1} h_{\alpha_i} \right] \qquad (3.29)$$

where, u is an arbitrary number of subintervals of [a, b], i=1,2,……u-1, $a + b = 0$, $\Delta\alpha = \frac{b-a}{u}$ and $\alpha_i = a + i.\Delta\alpha$. The Equation 3.29 can further be written as in Equation 3.30.

$$\int_a^b h_\alpha d\alpha \cong \frac{b-a}{2u}\left(h_a + 2h_{a+\Delta\alpha} + 2h_{a+2\Delta\alpha} + \cdots + 2h_{b-2\Delta\alpha} + 2h_{b-\Delta\alpha} + h_b\right) \tag{3.30}$$

By substituting the Equation 3.30 into Equation 3.28 one obtains a new representation of $\bar{h}_\alpha$ as in Equation 3.31.

$$\bar{h}_\alpha = \frac{1}{2u}\left(h_a + 2h_{a+\Delta\alpha} + 2h_{a+2\Delta\alpha} + \cdots + 2h_{b-2\Delta\alpha} + 2h_{b-\Delta\alpha} + h_b\right) \tag{3.31}$$

3.4.5.4 Ranking of Alternatives

Alternatives are ranked based on their respective mean values calculated in the previous subsection.

3.4.5.5 Sensitivity Analysis

The sensitivity analysis of a decision making problem consists in determining the set of criteria for which the smallest change of their weights will impact the ranking order of alternatives (Wallace 1998). As the sum of criteria weights is always equal to 1, a change of one criterion weight will lead to a change of other criteria weights. Let us assume the weight w_i is changed *into* $w_i^{'}$, the change of another criterion weight w_k into $w_k^{'}$ is expressed as in Equation 3.32.

$$w_k^{'} = \frac{1-w_i^{'}}{1-w_i} \cdot w_k \tag{3.32}$$

3.4.6 ELECTRE

The Elimination and Choice Translating Reality (ELECTRE) method was first introduced by Benayoun et al. (1966). It consists of implementing pairwise comparisons between alternatives based on their respective performances against different criteria of a given decision making problem. The pairwise comparison between two alternatives determines the outranking relationship between them. The outranking relationship between two alternatives A_r and A_s describes the dominance of the alternative A_r over the alternative A_s (Uysal and Yavuz 2014). The outranking relationships between alternatives are determined through the analysis of the concordance and discordance indexes. The former are defined as the set of evidences that the alternative A_r dominates the alternative A_s while the later provides proof that A_s is dominated by A_r (Ermatita, Hartati, Wardoyo and Harjoko 2011).

The ELECTRE algorithm consists of the following steps:

- Compute the normalized decision matrix,
- Compute the weighted normalized decision matrix,
- Determine the concordance and discordance sets,
- Compute the concordance and discordance matrices,
- Compute the concordance and discordance dominance matrices,
- Compute the aggregate dominance matrix and
- Ranking of alternatives.

3.4.6.1 Compute the Normalized and Weighted Normalized Decision Matrices

The normalized decision matrix (R) and weighted normalized decision matrix (V) in ELECTRE are determined the same way as in TOPSIS with Equations 3.12 and 3.13, respectively.

3.4.6.2 Determination of the Sets of Concordances and Discordances

The set of concordances Con_{rs} between two alternatives A_r and A_s includes all criteria where A_r is more effective than A_s in the normalised weighted matrix V. It is defined as in Equation 3.33.

$$Con_{sr} = \{j | v_{sj} \geq v_{rj}\} \quad j = 1,2,3\cdots, N \tag{3.33}$$

The set of discordances Dis_{rs} between two alternatives A_r and A_s is the reverse of Con_{sr}; it includes the set of criteria where A_r is less effective (performing) than A_s in the normalised weighted matrix V. It is defined as in Equation 3.34.

$$Dis_{sr} = \{j | v_{sj} < v_{rj}\} \quad 1 \leq j \leq N \tag{3.34}$$

3.4.6.3 Compute the Concordance and Discordance Matrices

The concordance matrix C_{sr} is a square matrix (M×M) where each c_{sr} (for $s \neq r$) element is the concordance index of an A_r compared to an alternative A_s. The concordance index of two alternatives A_r and A_s is the sum of weights of criteria of their set of concordances; it is defined as in Equation 3.35.

$$c_{sr} = \sum_{j \in Con_{sr}} w_j \quad 1 \leq j \leq N \tag{3.35}$$

Using the concordance indexes, the concordance matrix is defined in Equation 3.36.

$$C_{sr} = \begin{bmatrix} - & c_{12} & c_{13} & \cdots & c_{1M} \\ c_{21} & - & c_{23} & \cdots & c_{2M} \\ c_{31} & c_{32} & - & \ddots & c_{3M} \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ c_{M1} & c_{M2} & c_{M3} & \cdots & - \end{bmatrix} \tag{3.36}$$

The discordance matrix D_{sr} is also a square matrix (M×M) where each d_{sr} element is the discordance index of an A_r compared to an alternative A_s. The discordance index of two alternatives A_r and A_s is computed from the weighted normalized decision matrix V as in Equation 3.37.

$$d_{sr} = \frac{\max_{j \in Dis_{sr}} \left| v_{sj} - v_{rj} \right|}{\max_{j} \left| v_{sj} - v_{rj} \right|} \tag{3.37}$$

The discordance matrix that includes the discordance indexes is represented in Equation (3.38).

$$D_{sr} = \begin{bmatrix} - & d_{12} & d_{13} & \cdots & d_{1M} \\ d_{21} & - & d_{23} & \cdots & d_{2M} \\ d_{31} & d_{32} & - & \ddots & d_{3M} \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ d_{M1} & d_{M2} & d_{M3} & \cdots & - \end{bmatrix} \tag{3.38}$$

One can notice that in the concordance and discordance matrices for $s=r$ the value is not specified; this is due to the fact that the sets of concordance and discordance of an alternative A_r are empty when A_r is compared to itself; therefore, its concordance and discordance indexes cannot be determined.

3.4.6.4 Compute the Concordance and Discordance Dominance Matrices

The concordance dominance matrix CD is a square matrix (M×M) where each element cd_{sr} is equal to 1 or 0 representing the Boolean outcome of the comparison between the concordance indexes of two alternatives A_r and A_s and the threshold value $\bar{c}$ of the concordance matrix C_{sr}. The threshold $\bar{c}$ is defined in Equation 3.39.

$$\bar{c} = \frac{\sum_{s=1}^{M}\sum_{r=1}^{M} c_{sr}}{M(M-1)} \tag{3.39}$$

Each element cd_{sr} of the CD matrix is then obtained by comparing the elements c_{sr} of the concordance matrix C_{sr} (Equation 3.36) to the threshold as in Equation 3.40.

$$\begin{cases} cd_{sr} = 1 \Rightarrow c_{sr} \geq \bar{c} \\ cd_{sr} = 0 \Rightarrow c_{sr} < \bar{c} \end{cases} \tag{3.40}$$

A similar process as that described above for the concordance dominance matrix is applied to obtain the discordance dominance matrix. The threshold value $\bar{d}$ of the discordance matrix is calculated as in Equation 3.41.

$$\bar{d} = \frac{\sum_{s=1}^{M}\sum_{r=1}^{M} d_{sr}}{M(M-1)} \tag{3.41}$$

Each element dd_{sr} of the DD matrix is then obtained by comparing the elements d_{sr} of the discordance matrix D_{sr} (Equation 3.38) to the threshold as in Equation 3.42.

$$\begin{cases} dd_{sr} = 1 \Rightarrow d_{sr} \geq \bar{d} \\ dd_{sr} = 0 \Rightarrow d_{sr} < \bar{d} \end{cases} \tag{3.42}$$

3.4.6.5 Compute the Aggregate Dominance Matrix

Each element e_{sr} of the aggregate dominance matrix E is obtained by multiplying each element cc_{sr} of the CD matrix (Equation 3.36) to the element dd_{sr} of the DD matrix (Equation 3.38) as in Equation 3.43.

$$e_{sr} = cc_{sr} \cdot dd_{sr} \tag{3.43}$$

The elements e_{sr} in Equation 3.43 form the matrix E as in Equation 3.44.

$$E = \begin{bmatrix} - & e_{12} & e_{13} & \cdots & e_{1M} \\ e_{21} & - & e_{23} & \cdots & e_{2M} \\ e_{31} & d_{32} & - & \ddots & e_{3M} \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ e_{M1} & e_{M2} & e_{M3} & \cdots & - \end{bmatrix} \tag{3.44}$$

Similar to the concordance and discordance matrices, the values e_{sr} (for s=r) of the matrix E are not determined. In fact, the aggregate dominance matrix is constructed from the concordance and discordance dominance matrices which have undefined values in their diagonals.

3.4.6.6 Ranking of Alternatives

Alternatives are ranked based on the dominance matrix E (Equation 3.42).

3.5 CONCLUSION

The ontology acquisition technique used in this study was presented in this chapter. Thereafter, the algorithms and programming environment used to compute the ontologies' complexity metrics were described. The 5 MADM methods for ranking the collected ontologies based on their complexity metrics was presented in detail. The next chapter presents the framework for analysing the complexity of ontologies.

CHAPTER 4: FRAMEWORK FOR ANALYSING ONTOLOGY COMPLEXITY

4.1 INTRODUCTION

This chapter presents the framework designed for analysing the complexity level of ontologies. It provides a detailed description of the different phases of the framework along with an outline of the way they can be implemented. The first phase consists of the acquisition or collection of ontologies of a given domain. Thereafter, the complexity metrics are computed for each of the collected ontologies in the second phase. The values computed in the second phase are used as inputs to multi-attributes decision method (MADM) algorithms in the third phase to provide to each of the collected ontologies a score representing its level of complexity. The scores obtained in the third phase from different MADMs are analysed in the last phase in order to validate the framework.

4.2 FRAMEWORK OVERVIEW

The framework consists of four phases as presented in Figure 4.1. These phases are: Ontology acquisition, complexity metric computation, Ontology ranking and validation. The phases of the framework are explained in detail in the next subsections.

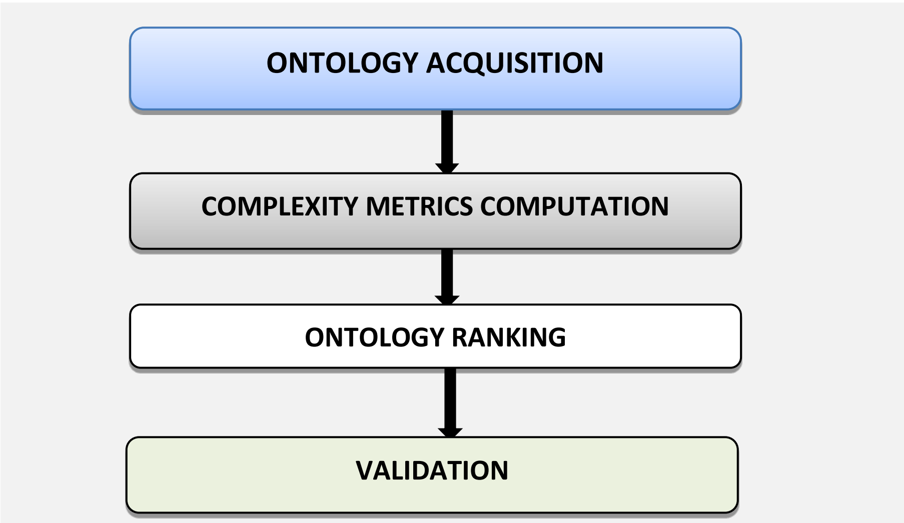

Figure 4.1 Framework for Analysing Ontology Complexity

4.2.1 Ontology acquisition

This phase consists of collecting ontologies in RDF or OWL formats from the Internet through the use of ontology search engines and repositories. There are many ontology search engines which enable easy access to RDF and OWL ontologies on the Web. Popular ontology search engines are Watson (D'Aquin and Motta 2011) and Swoogle (Ding 2005), whereas, LODE (Peroni et al. 2012) and BioPortal (Salvadores, Alexander, Musen and Noy 2013) are examples of ontology repositories. In this study, biomedical ontologies are collected from the BioPortal repository.

4.2.2 Complexity Metrics Computation

Ontologies collected in the first phase are used as inputs of this phase. For each ontology, a number of complexity metrics are computed including: the entropy of ontology graph, depth of inheritance, average number of paths per class, tree impurity, class richness, relationship richness, average path length for a given class and average path length of ontology. These complexity metrics are computed through the implementation of different algorithms presented in Chapter 3. These algorithms can be implemented using Semantic Web platforms such as Jena API (Application Programming Interface), Allegrograph, Sesame or Minerva (Ramanujam et al. 2009; Fan et al. 2010; Zhou 2010). Jena API is used in this study.

4.2.3 Ontology Ranking

The framework in Figure 4.1 implements a set of MADMs to rank the ontologies based on their complexity metrics computed in the second phase. Several MADMs methods have been developed to help solve problems involving ranking a set of alternatives based on their performances over a set of criteria. Examples of these MADMs include: Weighted Sum Method (WSM), Weighted Product Method (WPM), Aggregated Indices Randomization Method (AIRM), Best Worst Method (BWM), Inner Product Vector (IPV), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), Analytical Hierarchy Process (AHP),Weighted Linear Combination Ranking Technique (WLCRT), and Elimination and Choice Translating Reality (ELECTRE) (Benayoun et al. 1966; Fishburn 1967; Miller et al. 1969; Saaty 1977; Triantaphyllou 1998; Chou 2013).

4.2.4 Validation

This phase is used to evaluate the ranking from the different MADMs implemented in phase 3. These results are used as inputs to the validation phase. The evaluation process consists in the determination of the impact of each metric on the overall complexity level of a given ontology. These results are validated by comparing them to the descriptions and roles played by each of the metrics in influencing the complexity of an ontology by taking into account the weight assigned to each of the metrics.

4.3 RELATED WORKS

Ontology ranking has been the focus of several research works. The OS_RANK platform is presented by Yu, Li, Chen and Cao (2007). This platform receives from the user a query where each term is an ontology class with a weight. This query is then submitted to an ontology repository or search engines such as Swoogle which returns a set of ontologies containing the query terms. The OS_RANK platform then uses a specific algorithm that ranks these ontologies based on the queries' terms.

Groza, Dragoste, Sincai and Jimborean (2014) implemented the Analytic Hierarchy Process (AHP) a MADM method to rank ontologies based on user preferences. The user preferences are expressed in terms of coverage, size, cohesion, consistency and language expressivity of the ontology. Another study by Tartir and Arpinar (2007) presented OntoQA, an ontology quality analysis and ranking platform that evaluates ontologies of a domain and ranks them based on their schemas and instances. OntoQA ranks ontologies by implementing the Weighted Sum Method (WSM) method.

In Collins and Clark (2004), a model for evaluating ontology schemas via two quantifiable and non-quantifiable features is proposed. The technique consists of crawling the web to search for ontologies and store them locally; thereafter, information such as the domain and weights of user defined metrics are used to return the suitable ontology. The authors further implemented the simple additive or WSM method to compute the score and rank the evaluated ontologies.

Alani and Brewster (2006) presented AKTiveRank, a system used to rank ontologies based on a set of metrics, namely, class match, density, semantic similarity and betweenness. These metrics are used to determine the degree of representation of concepts in an ontology. Similar to OntoQa and OS_RANK, AKTiveRank is connected to an ontology search engine in order to collect ontologies that match user's query and rank them using the WSM. None of the abovementioned studies has focused on analysing and ranking ontologies based on their complexity.

4.4 CONCLUSION

In this chapter, a framework designed to rank ontologies based on their complexity was presented. The framework consists of the following phases: Ontology acquisition, complexity metric computation, Ontology ranking and validation. The Chapter presented each phase of the framework in terms of existing methods and tools needed to implement it. Furthermore, the related works done in the field of ontologies analysis and ranking were discussed. However, none of the presented studies addressed the analysis and ranking of ontologies based on their complexity. The next chapter presents the experimental results of the application of the proposed framework.

CHAPTER 5: EXPERIMENTS AND DISCUSSION

5.1 INTRODUCTION

This chapter presents and discusses the experimental results of the application of the framework for analysing the complexity of the ontologies drawn in Chapter 4. The dataset and the programming environment as well as the primitive complexity metrics of the ontologies in the dataset are presented. Thereafter, the results of the application of the algorithms developed in Chapter 3, Section 3.3 and the 5 MADMs presented in Chapter 3, Section 3.4 are presented and discussed.

5.2 DATASET

The dataset constituted 100 ontologies downloaded from the BioPortal Repository. These ontologies are listed in Tables 5.1 and 5.2 and are the semantic modelling of different branches of the biomedical domain. They include:

- Ontologies of different kinds of diseases and their impact on human and animal bodies – Examples are the Alzheimer disease ontology (O_2 in Table 5.1), HIV ontology (O_{73} in Table 5.1) and Dengue Fever ontology (O_7 in Table 1).

Table 5.1 List of Biomedical Ontologies in the Dataset- Part I

Index	Ontology Name	Index	Ontology Name
$\mathbf{O_1}$	Information Consent Ontology	O_{26}	Non-coding RNA
$\mathbf{O_2}$	Alzheimer's Disease Ontology	O_{27}	Semantic Science Ontology
$\mathbf{O_3}$	Bone dysplasia Ontology	O_{28}	Statistic Ontology
$\mathbf{O_4}$	Cigarette Smoke Exposure Ontology	O_{29}	Neural Electromagnetic Ontology
$\mathbf{O_5}$	Ontology of vaccine advert events	O_{30}	New Born Ontology
$\mathbf{O_6}$	Dermatology Lexicon	O_{31}	Parkinson Disease Ontology
$\mathbf{O_7}$	Dengue Fever Ontology	O_{32}	Animal trait ontology
$\mathbf{O_8}$	Galen Ontology	O_{33}	Ontology of Pneumology
$\mathbf{O_9}$	Human Dermatological Ontology Disease	O_{34}	Metagenome and Microbiology Ontology
$\mathbf{O_{10}}$	Human Interaction Network Ontology	O_{35}	Human Physiology simulation ontology
$\mathbf{O_{11}}$	Natural Products Ontology	O_{36}	Sleep Domain Ontology
$\mathbf{O_{12}}$	NCI Thesaurus	O_{37}	The Drug-Drug Interaction Ontology

O_{13}	Ontology of Adverse Events	O_{38}	Hymenoptera Anatomy Ontology
O_{14}	Ontology of drug neuropathy adverse event	O_{39}	Congenital Health Defects
O_{15}	Orphanet Rare Disease Ontology	O_{40}	Environment ontology for livestock
O_{16}	Uber Anatomy Ontology	O_{41}	Phenotype Quality Ontology
O_{17}	Vaccine Ontology	O_{42}	Human dermatological disease Ontology
O_{18}	Experimental Factor Ontology	O_{43}	Cognitive Atlas Ontology
O_{19}	Human Disease Ontology	O_{44}	Cell type ontology
O_{20}	Cell Ontology	O_{45}	Ontology of physics for biology
O_{21}	Human Phenotype Ontology	O_{46}	Ontology of MicroRNA Target
O_{22}	Chemical Entities of Biological Interest	O_{47}	Mass Spectrometry
O_{23}	Diabetes Ontology	O_{48}	Adult mouse brain
O_{24}	Nano particle Ontology	O_{49}	Ontology of biological and clinical statistic
O_{25}	Pathogenic diseases	O_{50}	Radio oncology ontology

- Ontologies of human and animal anatomy - These ontologies encompass the vertebrate skeletal ontology (O_{51} in Table 5.2) and anatomical entity ontology (O_{71} in Table 5.2).
- Ontologies of treatment products and their effects on the human body - Examples of these ontologies include the vaccine ontology (O_{17} in Table 5.1), the ontology of adverse events (O_{13} in Table 5.1) and the Natural products ontology (O_{11} in Table 5.1).

Table 5.2 List of Biomedical Ontologies in the Dataset – Part II

Index	Ontology Name	Index	Ontology Name
O_{51}	Vertebrate Skeletal Ontology	O_{76}	Eagle resource research
O_{52}	BioAssay Ontology	O_{77}	Plant experimental assay Ontology
O_{53}	Emotion Ontology	O_{78}	Ontology of Drug Neuropathy adverse events
O_{54}	Neuroscience Ontology	O_{79}	Neural-Immune Gene Ontology
O_{55}	Neuroscience Information Ontology	O_{80}	Kinetic simulation algorithm ontology
O_{56}	Ontology of genetic interval	O_{81}	Chemical Information Ontology
O_{57}	Population and Community Ontology	O_{82}	Sequence phenotype ontology
O_{58}	Beta Cell Genomics Ontology	O_{83}	Disease core rare disease Ontology
O_{59}	Enano Mapper Ontology	O_{84}	Drug Interaction Knowledge Base Ontology
O_{60}	Experimental Factor Ontology	O_{85}	Cell line Ontology

O_{61}	Immuno-genetics Ontology	O_{86}	Breast Cancer Ontology
O_{62}	NanoParticle Ontology	O_{87}	Multiple Sclerosis Ontology
O_{63}	Brain Region Ontology	O_{88}	Autism spectrum ontology
O_{64}	Mental Functioning ontology	O_{89}	Infectious Disease Ontology
O_{65}	Clinical Measurement Ontology	O_{90}	Translational medicine ontology
O_{66}	Fission Yeast Phenotype Ontology	O_{91}	Ecosystem ontology
O_{67}	Adult Brain Ontology	O_{92}	Ontology of alternative medicine
O_{68}	Clinical Trials Ontology	O_{93}	Family Health History Ontology
O_{69}	Fanconi Anemia Ontology	O_{94}	Symptom Ontology
O_{70}	Medical image simulation	O_{95}	Cancer Management and research ontology
O_{71}	Anatomical entity Ontology	O_{96}	Biomedical Resource Ontology
O_{72}	Single-Nucleotide Polymorphism Ontology	O_{97}	Growth medium ontology
O_{73}	HIV Ontology	O_{98}	Epidemiology Ontology
O_{74}	Cardiac Electrophysiology Ontology	O_{99}	Ontology of clinical research
O_{75}	Flora phenotype	O_{100}	Mental State Assessment

- Ontologies of organization of molecules and proteins and their different processes in the human and animal bodies – Examples are: cell ontology (O_{20} in Table 5.1), sequence phenotype ontology (O_{82} in Table 5.2) and the Non-coding RNA (O_{26} in Table 5.1).
- Ontologies of cancer and treatment methods - Examples of these include the Breast cancer ontology (O_{86} in Table 5.2), cancer management and research ontology (O_{95} in Table 5.2) and the radio oncology ontology (O_{50} in Table 5.1).

5.3 SOFTWARE ENVIRONMENT

The experiments were carried out on a computer with the following characteristics: 64-bit Genuine Intel (R) Celeron (R) CPU 847, Windows 8 release preview, 2 GB RAM and 300 GB hard drive. The algorithms for computing the complexity metrics were implemented in Java Jena API (McBride 2001) configured in Eclipse Integrated Development Environment (IDE) Version 4.2.

5.4 CALCULATION AND ANALYSIS OF COMPLEXITY METRICS

5.4.1 Calculation of Primitives Metrics of Ontology

In order to compute the advanced complexity metrics for all the ontologies in the dataset, it was necessary to determine the basic semantic characteristics of these ontologies such as the number of classes, properties and instances. To this end, appropriate data structures and programmes were designed and implemented in Java Jena API. Figures 5.1, 5.2 and 5.3 depict the charts of the basic semantic characteristics of the biomedical ontologies in the dataset including the number of classes (Figure 5.1), properties (Figure 5.2) and instances (Figure 5.3). These characteristics appear in Figures 5.1, 5.2 and 5.3 as pairs of values in the form x.y. The value x represents the index of the ontology in Table 5.1 or Table 5.2 and the value y, either the number of concepts, properties or instances of the ontology O_x.

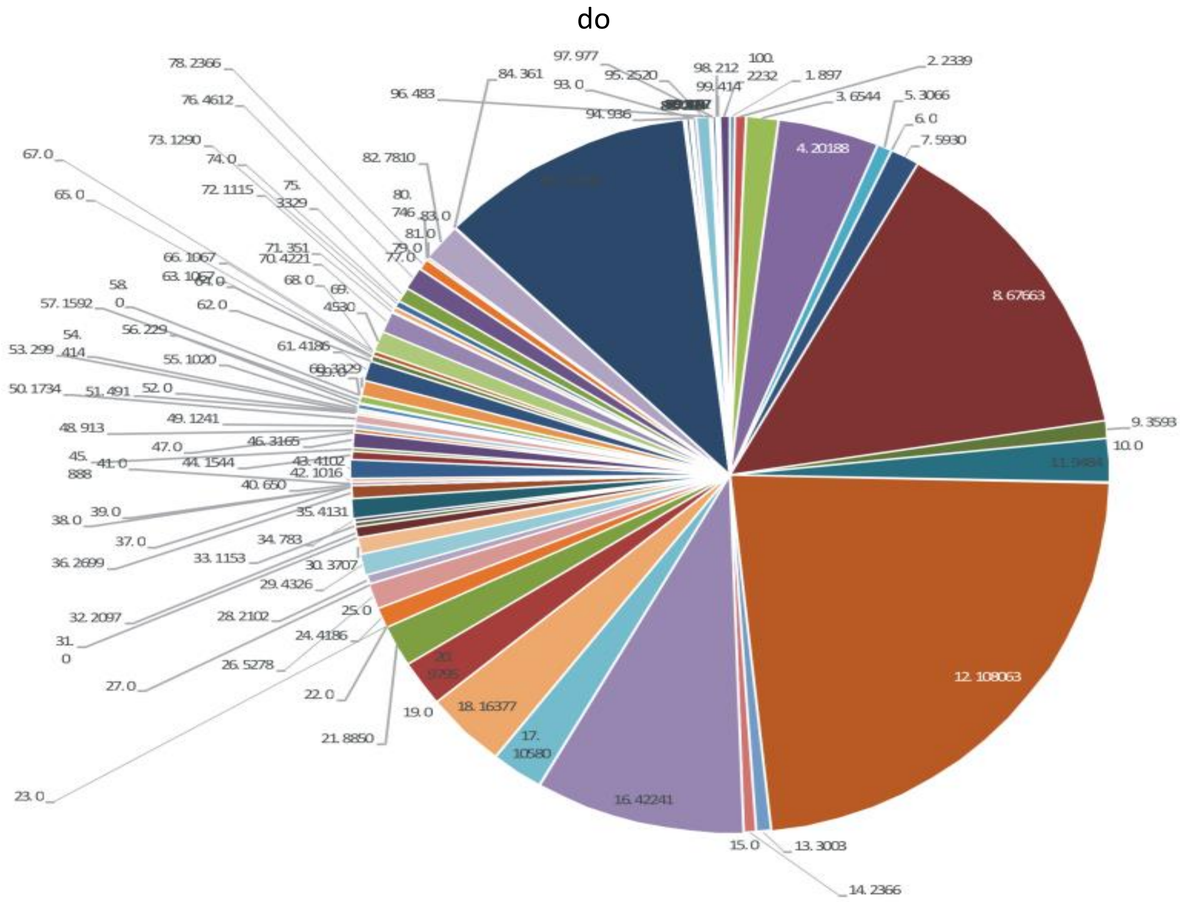

Figure 5.1 Number of Concepts in the Biomedical Ontologies in the Dataset

For instance, the largest bar in the chart in Figure 5.1 corresponds to the pair 12.108063; this means the ontology O_{12} in Table 1, that is, the National Cancer Institute (NCI) Thesaurus has

108063 classes. Similarly, the pair 78.2366 at the top left of Figure 5.1 means that the ontology O_{78} in Table 5.1, that is, the Ontology of Drug Neuropathy adverse events has 2366 classes. One can notice that the number of classes in some pairs of values in Figure 5.1 is zero; these pairs correspond to 30 ontologies that contained errors in their codes and could not be processed by the Java programme. For all these ontologies with incorrect codes, the corresponding pairs of values for the number of classes, properties and instances is zero in Figures 5.1, 5.2 and 5.3.

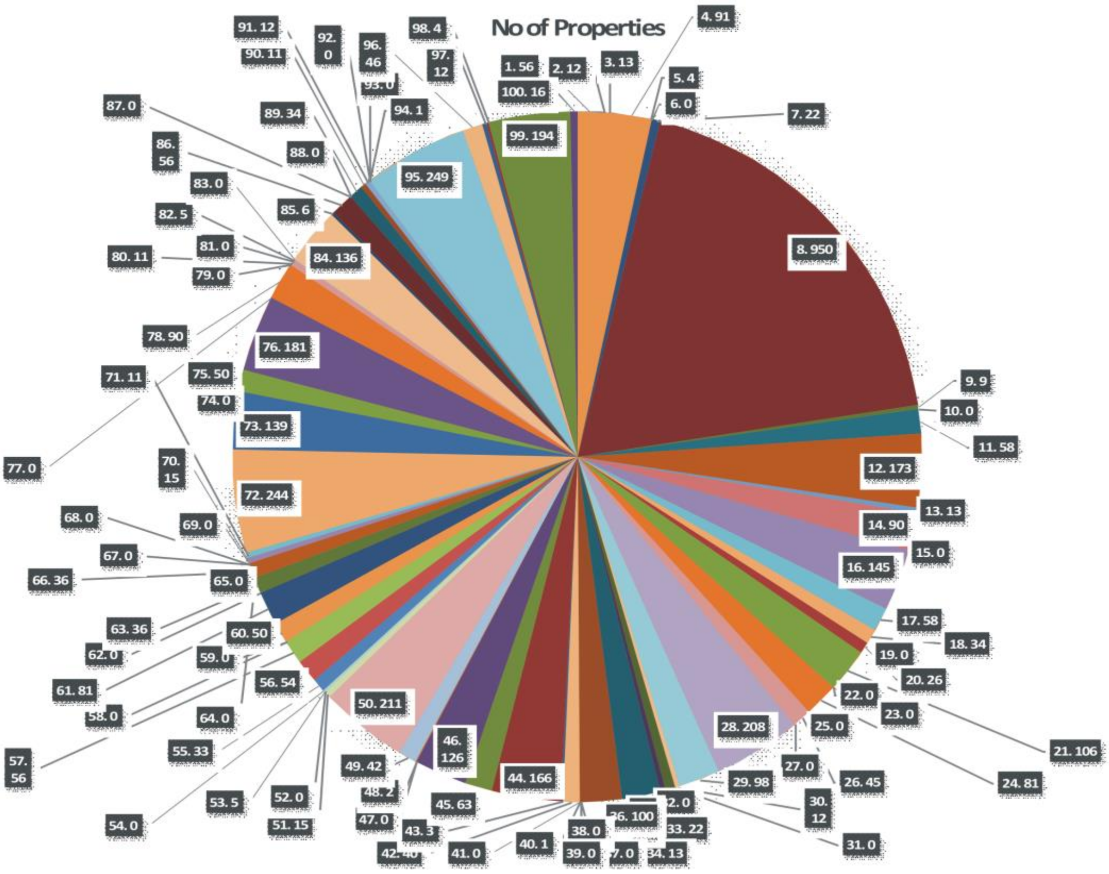

Figure 5.2 Number of Properties of Biomedical Ontologies in the Dataset

Figure 5.3 depicts the chart of the number of instances in the biomedical ontologies in the dataset. The ontology with the most instances is O_{11} in Table 5.1, that is, the Natural Products Ontology with 22012 instances, followed by the NCI Thesaurus (O_{12} in Table 5.1) with 4141 instances.

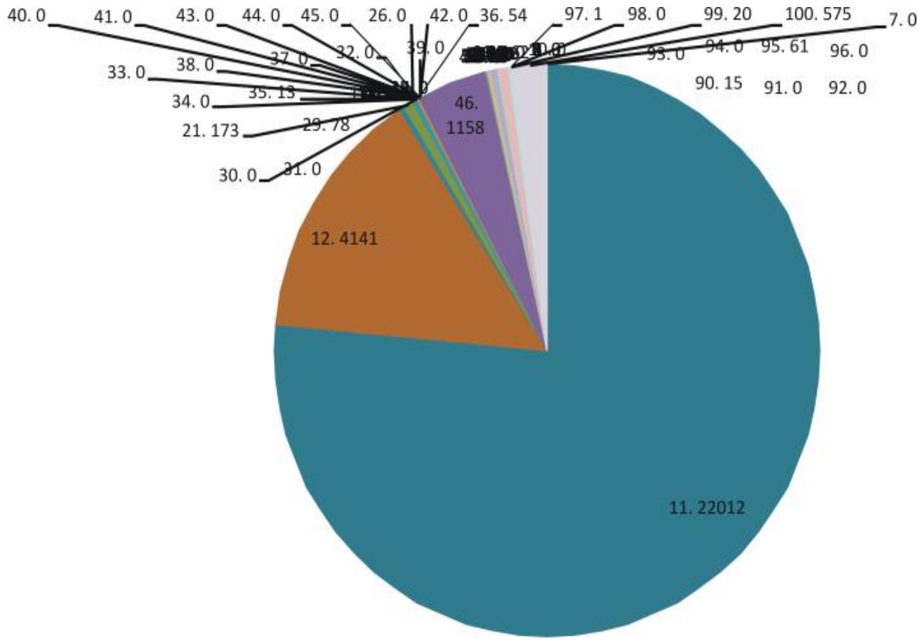

Figure 5.3 Number of the Instances of Biomedical Ontologies in the Dataset

Overall, Figure 5.3 shows that the majority of selected ontologies in the BioPortal as datasets for this study had a lower number of instances.

5.4.2 Calculation and Discussion of the Advanced Complexity Metrics of Ontologies in the Dataset

Let us recall that the advanced complexity metrics of ontologies discussed in detail in Chapter 2, Subsection 2.4.2.3, include: depth of inheritance (DIP), size of the vocabulary (SOV), entropy of ontology graphs (EOG), the average part length (APL) and average number of paths per class (ANP), the tree impurity (TIP), relationship richness (RR) and class richness (CR). Amongst the 100 ontologies in the dataset (Tables 5.1 and 5.2), only the codes of 70 were successfully parsed in Jena API to enable the calculation of their advanced complexity metrics with the algorithms presented in Chapter 3, Section 3.3. The following subsections analyse and discuss the advanced complexity metrics of the 70 ontologies that were successfully processed (listed in Appendix A).

5.4.2.1 Size of the Vocabulary

Figure 5.4 presents the results of the measurement of the Size of the Vocabulary (SOV) for ontologies in the dataset. These results are grouped into 8 ranges from the range of ontologies with a SOV less than 1k (i.e. 1000) to the range of the ones with SOV >100k (i.e. 100000).

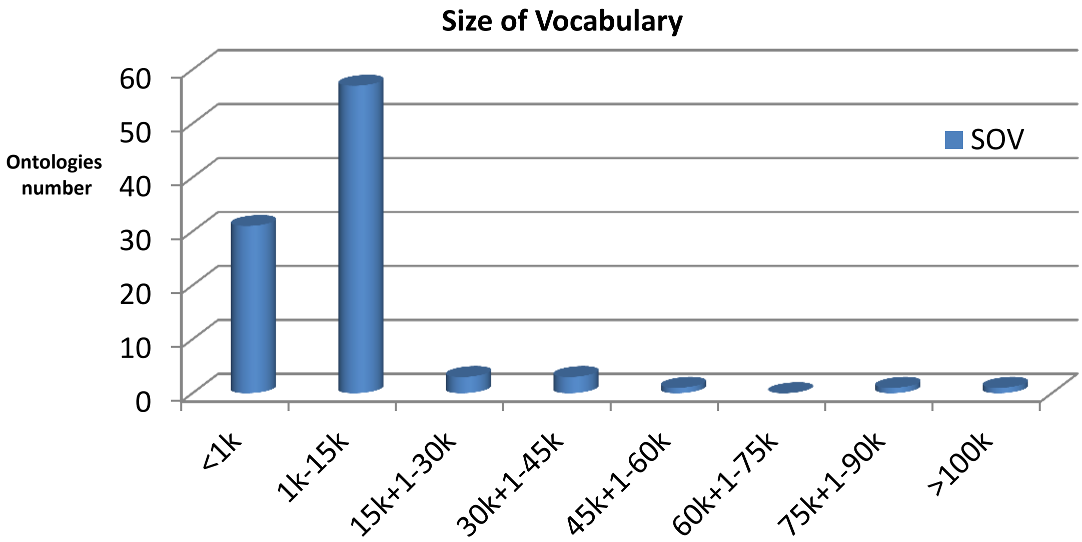

Figure 5.4 Size of Vocabulary

The majority (56%) of the ontologies in the dataset have a SOV between 1000 and 15000, followed by those with a SOV less than 1000 (31%); 5% of ontologies in the dataset have a SOV between 15000 to 30000 and 2% a SOV of more 100000. These results indicate that the majority of ontologies in the dataset are constituted of thousands or tens of thousands of components. Then, it would be beneficial for semantic web developers in the biomedical domain to consider the reuse of these larger ontologies (Uber Anatomy ontology, O_{16}, SOV=42386), Vaccine Ontology, O_{17}, SOV=10706)) rather than trying to build new related ontologies *de novo*. The SOV of these ontologies also suggests that they would require a larger amount of time and effort to build (Zhang et al. 2010).

5.4.2.2 Average path length of ontology and Average Number of Paths per Concept

Figure 5.5 presents a joint analysis of the average path length of the ontology and the average number of paths per concept or class (ρ). The values of these 2 metrics for all the ontologies in the dataset are grouped into 11 ranges as in Figure 5.5. Figure 5.5 shows that a considerable proportion of the ontologies in the dataset (36%) have a ρ value less than 5; a larger number of these ontologies (37%) have a ρ between 6 and 15; 6% of ontologies in the dataset have a ρ between 36 and 45. A smaller number of ontologies (4%) have ρ in one of the following ranges 16-25, 26-35, 46-55 and 66-75.

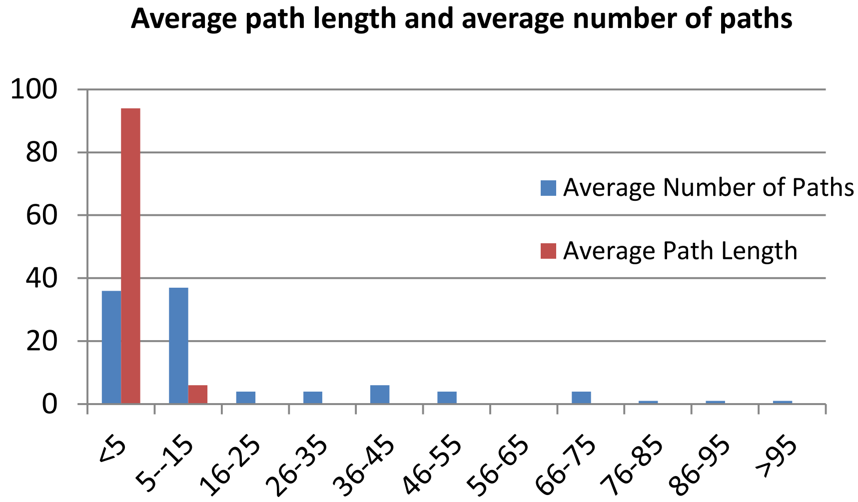

Figure 5.5 Average Path Length and Average Number of Paths

From the analysis of the ρ values for all the ontologies in the dataset, one can conclude that the majority of the ontologies in the dataset have multiple paths from the root class to given classes; this indicates that in most of these ontologies the inheritance relationships among the classes are intense and constitute a sign of higher complexity of these ontologies. Once more, building similar ones from scratch would require a lot of time and effort (Yang et al. 2006). Figure 5.5 also portraits that the majority of ontologies in the dataset (94%) have smaller $\overline{\Lambda}$ values (less than 5). This indicates that changes in a class in these ontologies would have a less impact on its sub-classes (Yang et al. 2006).

5.4.2.3 Entropy of the Ontology Graph or Inheritance Hierarchy

Figure 5.6 presents the chart of EOG for the ontologies in the dataset. The bars in the chart in Figure 5.6 represent the percentage of ontologies with EOG in the corresponding range of EOG values. Figure 5.6 depicts that many of the ontologies in the dataset have EOG between 2 and 2.499 (41%); followed by those with EOG in the range of 1.5 to 1.999.

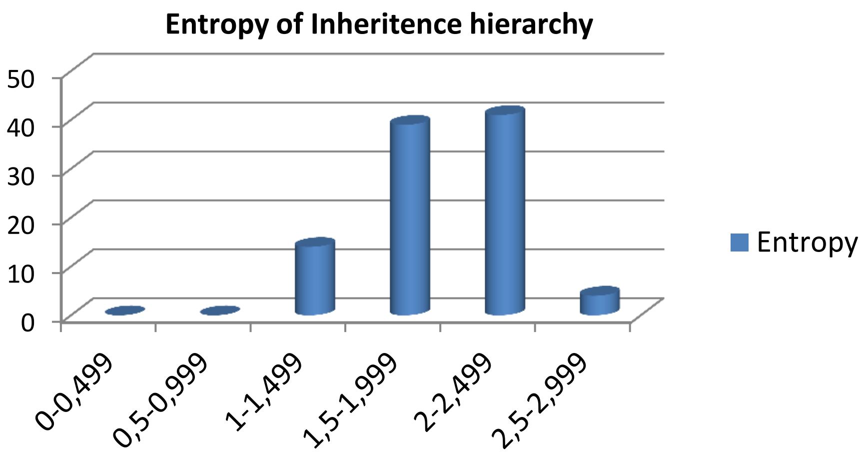

Figure 5.6 Entropy of Inheritance Hierarchy

A significant group of these ontologies have EOG between 1 and 1.499 (14%). A smaller number of the ontologies in the dataset have EOG close to zero. This indicates that the structures of the majority of ontologies in the dataset are less regular, which is a sign of higher complexity of these ontologies (Zhang et al. 2010).

5.4.2.4 Tree Impurity

Figure 5.7 presents results of the calculation of Tree Impurity (TIP) for all the ontologies in the dataset. These results are classified into 5 groups in Figure 5.7 based on the TIP values. It is shown in Figure 5.7 that an important number of ontologies in the dataset (44%) have TIP between 100 and 1000 (k); followed by those with TIP below 100 (21%). The remaining groups of ontologies have TIP in the ranges (1k+1) to 5k (1001 to 5000), (5k+1) to10k (5001 to 10000) and >10k (10000). These results suggest that the average number of subclass relations per class is low in these ontologies; this indicates that they can be easily reused and maintained (Zhang et al. 2010).

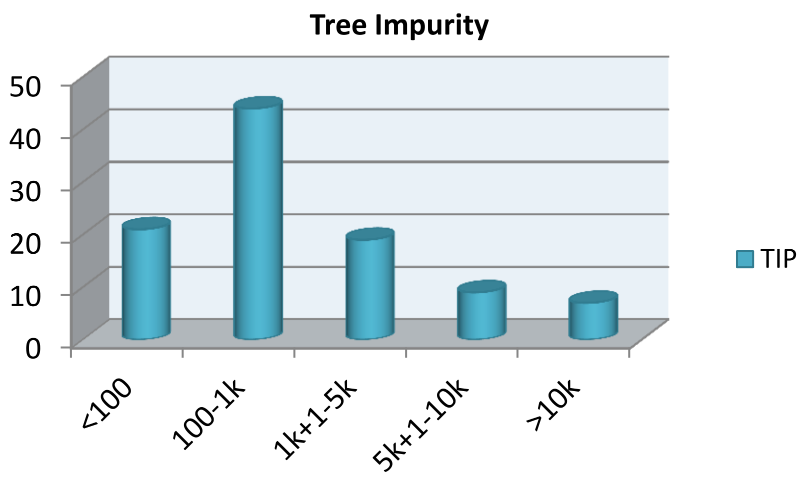

Figure 5.7 Tree Impurity

5.4.2.5 Relationship Richness

Figure 5.8 presents the results of a joint analysis of the relationship and class richness metrics. Figure 8 shows that 99% of the ontologies in the dataset have a RR between 0.5 and 0.74999 and all of them have CR values less than 0.25. This indicates that there is a balance between the number of SubClassOf and non-SubClassOf relationships and that most of the classes of these ontologies do not have instances.

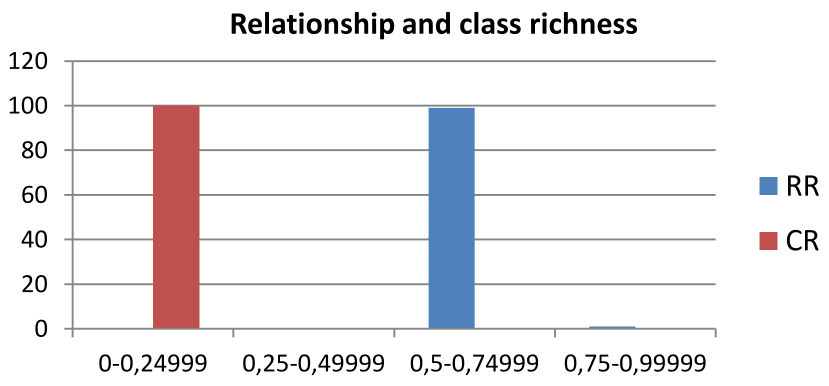

Figure 5.8 Relationship and Class Richness

5.5 PERFORMANCE ANALYSIS OF COMPLEXITY ALGORITHMS

The performance analysis consists in determining the execution time of the main algorithms designed to compute ontology complexity metrics including: *FINDPATHS*, *ENTROPY* and *RICHENESS* presented in Chapter 3, Subsection 3.3. This entails determining the asymptotic behaviour of the function *f (n)* of execution time of these 3 algorithms. The asymptotic behaviour of a function *f (n)* of an algorithm (or simply the asymptotic behaviour of an algorithm) refers to the growth of *f (n)* as *n* gets large with *n* representing the size of the input to the algorithm (Aurora and Barak 2009). The asymptotic behaviour of the 3 abovementioned algorithms is based on the Big-O notation which consists in considering only the variable *n* with its highest order and ignoring other low-order terms in *f (n)* (Aurora and Barak 2009).

5.5.1 FINDPATHS Algorithm

The FINDPATHS algorithm is presented in Chapter 3, Subsection 3.3.1. Based on the rule of thumb from the algorithm complexity theory related to the number of loops in an algorithm (Aurora and Barak 2009), the function *f(n)* of the FINDPATHS algorithm is O(n); this is due to the fact that FINDPATHS has three simple loops (not nested). Further, based on the rule of the worst-case or highest number of iterations of a loop (Aurora and Barak 2009) n is considered as the number of classes of the ontology evaluated.

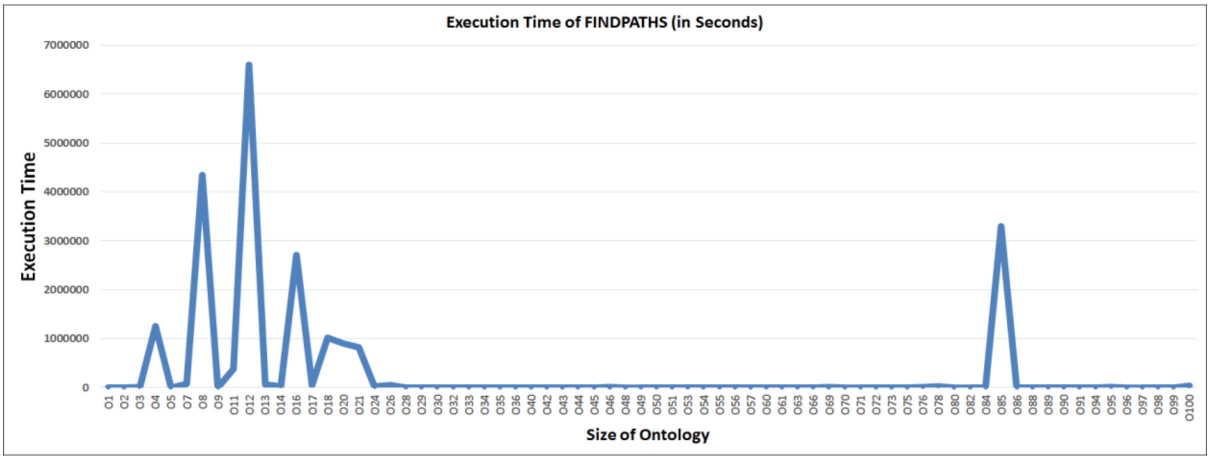

Figure 5.9 Execution Time of FINDPATHS Algorithm

Figure 5.9 presents the results of the execution time of the FINDPATHS algorithm on the dataset. The results in Figure 5.9 show that the execution time of FINDPATHS on the dataset is higher on the ontologies with large number of classes (e.g. O_5, O_9, O_{16} and O_{85}) and lower on the ontologies with low number of classes (e.g. O_{34}, O_{40}, O_{49} and O_{84}). This is an indication that the execution time of FINDPATHS depends on the number of classes of the ontology.

5.5.2 ENTROPY Algorithm

The ENTROPY algorithm (Chapter 3, Subsection 3.3.2) is mainly based on a loop that processes a list of integers where each value contained in it corresponds to the number of classes having their E(ci) equal to the position of the value in the list. Therefore, it can be deduced that the function f(n) of ENTROPY is O(n) with the worst-case corresponding to case where there are two classes ci and cj with E(ci) ≠ E(cj). Figure 5.10 presents the results of the execution time of the ENTROPY on the dataset.

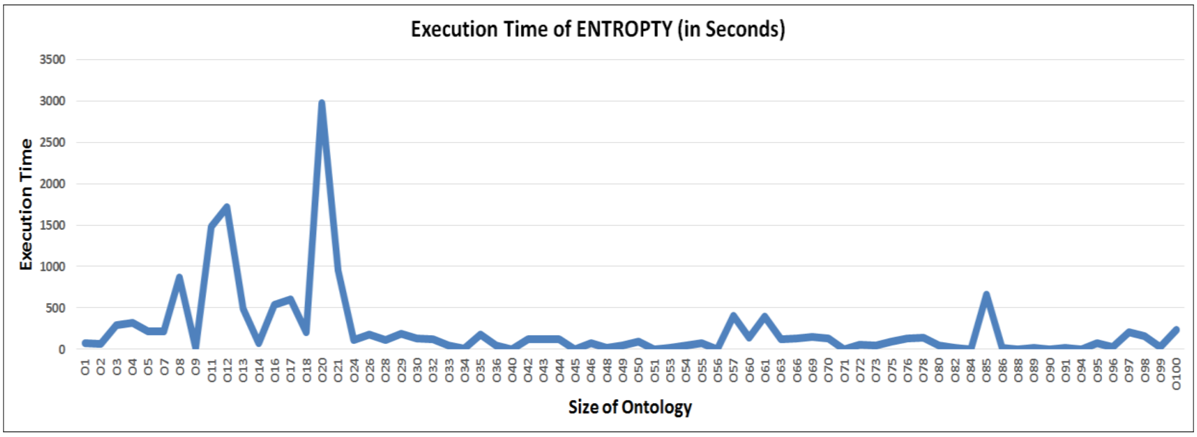

Figure 5.10 Execution Time of Entropy Algorithm

Once more the findings presented in Figure 5.10 tell that running ENTROPY over the ontologies with a big the number of classes (e.g. O_{11}, O_{12}, O_{20} and O_{21}) takes more time than running it on ontologies with smaller number of classes (e.g. O_{45}, O_{72}, O_{90} and O_{94}).

5.5.3 RICHNESS Algorithm

The execution of the RICHNESS algorithm (Chapter 3, Subsection 3.3.3) mainly relies on two loops which respectively process every class and every statement of the ontology. Therefore, the f(n) of RICHNESS is O(n) with the worst-case being when a high number of classes lead to a high number of statements in the ontologies.

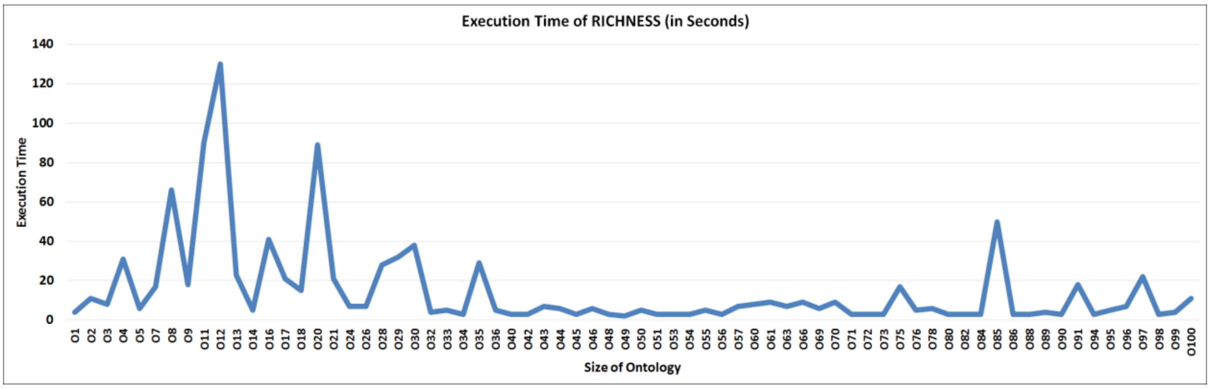

Figure 5.11 Execution Time of Richness

The results in the Figure 5.11 show that in most cases for the ontologies in the dataset with large number of classes (e.g. O_9, O_{12} and O_{20}), the execution time for RICHNESSES is higher than on the ontologies with lower number of classes (e.g. O_{57}, O_{84}, O_{89} and O_{90}).

5.6 RANKING OF ONTOLOGIES

The 5 MCDMs algorithms presented in Chapter 3, Section 3.4 were implemented to rank the ontologies in the dataset. The UML class diagram of the ranking system is presented in Figure 5.12.

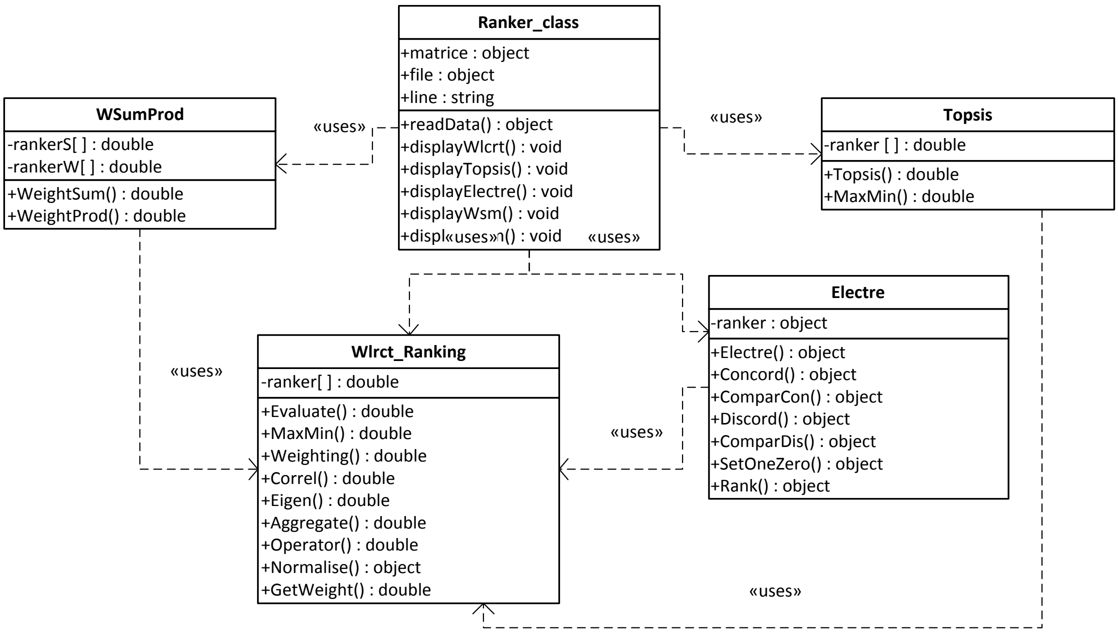

Figure 5.12 UML Class diagram of the ranking system

Figure 5.12 shows that each MCDM algorithm is implemented as a Java class. It can be noticed that the classes Electre, Topsis and WSumProd used the class WLCRT_Ranking; this is due to the fact that the set of criteria weights used in the experiments are determined with the WLCRT method.

5.6.1 Weighted Sum Method and Weighted Product Method

The Weighted Sum Method (WSM) and Weighted Product Method (WPM) scores are calculated by applying Equations 3.9 and 3.11 (Chapter 3, Subsections 3.4.2 and 3.4.3) on the normalised decision matrix in Appendix B. The resulting WSM and WPM scores are presented in Tables 5.3 and 5.4, respectively.

Table 5.3 WSM scores

Index	Score	Index	Score	Index	Score
O_{1}	0.3230362710809413	O_{33}	0.3444823485517981	O_{70}	0.24058983310515722
O_{2}	0.2671327933645708	O_{34}	0.19576479463104535	O_{71}	0.5670573138448306
O_{3}	0.24573247987856275	O_{35}	0.2903654864068279	O_{72}	0.27308260306155957
O_{4}	0.5985661331777506	O_{36}	0.38455515841796195	O_{73}	0.44692924693967384
O_{5}	0.2273228826552509	O_{40}	0.2527899321492712	O_{75}	0.2697443163352491
O_{7}	0.2940775654803628	O_{42}	0.295726214502752	O_{76}	0.30026628821298124
O_{8}	0.43407781032907206	O_{43}	0.20814958260575098	O_{78}	0.29515946605575377
O_{9}	0.2770973200484809	O_{44}	0.17898051940641166	O_{79}	0.5666026111435565
O_{11}	0.5576100761320892	O_{45}	0.25972019243957056	O_{80}	0.23840237869983538
O_{12}	0.3387474964601066	O_{46}	0.49598743583062743	O_{82}	0.39307879649755834
O_{13}	0.22724A856362657128	O_{48}	0.22861823904400236	O_{84}	0.28093679223313117
O_{14}	0.5683172631324995	O_{49}	0.3340510276571831	O_{85}	0.5597078199804438
O_{16}	0.5162263355663921	O_{50}	0.3038578597463658	O_{86}	0.27006016169776803
O_{17}	0.2652731139403928	O_{53}	0.2749129371583716	O_{88}	0.20313503048730233
O_{18}	0.3083602341616226	O_{54}	0.2287036735977184	O_{89}	0.2587149852926783
O_{20}	0.34298419496413257	O_{55}	0.3744515690102136	O_{90}	0.2812317182730927
O_{21}	0.3536017726880767	O_{56}	0.24418013030935593	O_{91}	0.23183015730429235
O_{24}	0.5483621339701298	O_{57}	0.34566338262773977	O_{94}	0.213434097426288
O_{26}	0.35467310181441203	O_{60}	0.5046755659996219	O_{95}	0.35701247766968
O_{28}	0.23845839614842806	O_{61}	0.3027232219270439	O_{96}	0.20635598859804904
O_{29}	0.5938837748636754	O_{63}	0.45061677981214115	O_{97}	0.19320817589200673
O_{30}	0.2254787679719866	O_{66}	0.2428301803568184	O_{98}	0.3179822143850816
O_{32}	0.25972010556937347	O_{69}	0.1894564023337689	O_{99}	0.224907854218235
				O_{100}	0.37399366084257024

The WSP scores in Table 5.3 resulted in the ranking of the ontologies in the dataset as in Figure 5.13. The ranking is provided in increasing order from 1 to 70.

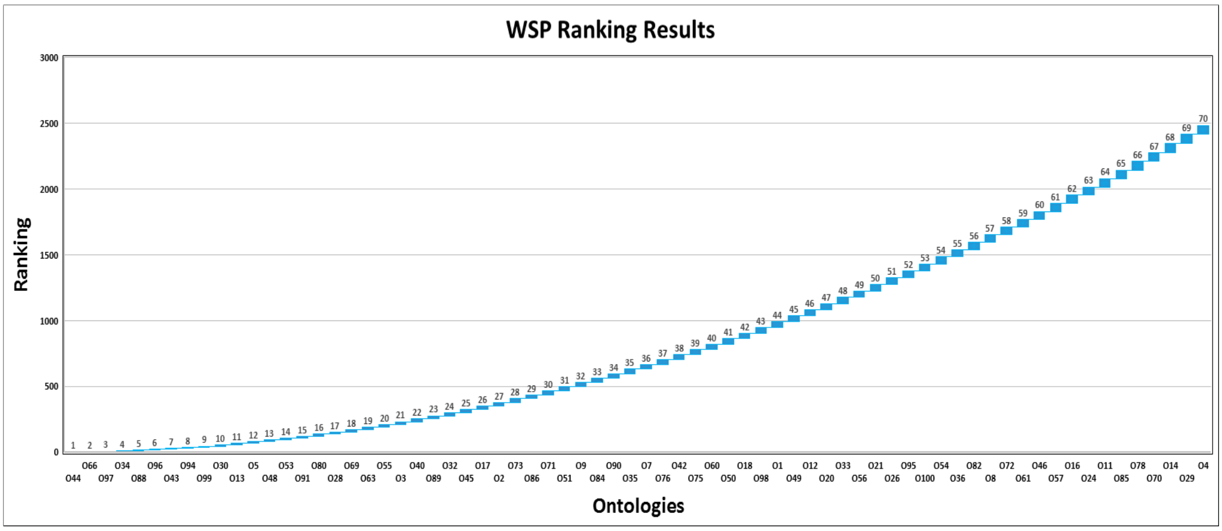

Figure 5.13 WSP Ranking Results

Due to the large number of ontologies (70) involved in the ranking in Figure 5.13, patterns of information are going to be looked at in three regions in the ranking, namely, the first, middle and last 10 positions; these are the ranges of positions: 1 to 10, 31 to 40 and 61 to 70.

The first 10 positions (1 to 10) in the ranking in Figure 5.13 are occupied by the ontologies including:

- O_{44} (Cell type ontology),
- O_{66} (Fission yeast phenotype ontology),
- O_{97} (Growth medium ontology, position 3),
- O_{34} (Metagenome and microbiology ontology),
- O_{88} (Autism spectrum ontology),
- O_{96} (Biomedical resource ontology),
- O_{43} (Cognitive atlas ontology),
- O_{94} (Symptom ontology),
- O_{99} (Ontology of clinical research) and
- O_{30} (New born ontology).

It was discussed in Chapter 2, Subsection 2.4.2.2 that ontologies with high values for the complexity metrics including: DIT, ANP, APL and TIP are highly complex. The analysis of the complexity metrics (Appendix A) of the abovementioned first 10 ontologies in the WSP ranking reveals that they have lower values for the DIT, ANP, APL and TIP as shown in Table 5.4.

Table 5.4 Complexity Metrics for the First 10 Ontologies in the WSM Ranking

	Ranges of Ontology Complexity Metrics			
	4 ≤ DIP ≤ 24	**1 ≤ ANP ≤ 133**	**1 ≤ APL ≤ 6**	**1 ≤ TIP ≤ 58741**
O_{44}	5	4	1	2
O_{66}	7	4	1	715
O_{97}	5	3	1	2
O_{34}	6	1	1	5
O_{88}	6	5	1	1
O_{96}	7	5	1	20
O_{43}	6	3	1	2507
O_{94}	7	3	1	98
O_{99}	6	3	1	257
O_{30}	4	3	1	1048

This is an indication that the first 10 ontologies in the WSM ranking are less complex compared to the rest of the dataset. This finding is supported by Figures 5.1 and 5.2 where these ontologies have low number of classes and properties. Furthermore, the low APL values of these ontologies indicates a small intensity of inheritance relationships among their concepts (Zhang et al. 2010). The last 10 positions (61 to 70) in the WSP ranking in Figure 5.13 are occupied by the ontologies including:

- O_{57} (Population and community ontology),
- O_{16} (Uber anatomy ontology),
- O_{24} (Nano particle ontology),

- O_{11} (Natural products ontology),
- O_{85} (Cell line ontology),
- O_{78} (Ontology of drug neuropathy adverse event),
- O_{70} (Medical image simulation),
- O_{14} (Ontology of drug neuropathy adverse events),
- O_{29} (Neural Electromagnetic ontology) and
- O_4 (Cigarette smoke exposure ontology).

The abovementioned last 10 ontologies in the WSM ranking in Figure 5.13 have higher values for the complexity metrics: DIT, ANP, APL and TIP as in Table 5.5; this indicates their high level of complexity.

Table 5.5 Complexity Metrics for the Last 10 Ontologies in the WSM Ranking

	Ranges of Ontology Complexity Metrics			
	4 ≤ DIP ≤ 24	**1 ≤ ANP ≤ 133**	**1 ≤ APL ≤ 6**	**1 ≤ TIP ≤ 58741**
O_{57}	18	1	1	907
O_{16}	16	71	3	30562
O_{24}	19	45	5	5176
O_{11}	19	44	5	2919
O_{85}	14	50	5	39915
O_{78}	16	13	2	781
O_{70}	9	7	2	455
O_{14}	22	49	6	1533
O_{29}	21	75	6	2644
O_4	21	75	6	122

The higher complexity of these ontologies is also evidenced in Figures 5.1 and 52 where they hold bigger number of classes and properties. The middle 10 positions in the WSM ranking in Figure 5.3 are occupied by the ontologies:

- O_{51} (Vertebrate Skeletal Ontology),
- O_{9} (Human Dermatological Ontology Disease),
- O_{84} (Drug Interaction Knowledge Base Ontology),
- O_{90} (Translational Medicine Ontology),
- O_{35} (Human Physiology Simulation Ontology),
- O_{7} (Dengue Fever Ontology),
- O_{76} (Eagle Resource Research),
- O_{42} (Human Dermatological Disease Ontology),
- O_{75} (Flora Phenotype Ontology) and
- O_{60} (Experimental Ontology).

Table 5.6 shows that the abovementioned middle 10 ontologies in the WSM ranking have higher values for the complexity metrics including DIT, ANP, APL and TIP, than those in the first 10 positions (Table 5.4); furthermore, these metrics are lower than that of the ontologies in the last 10 positions (Table 5.5). This finding suggests that the WSM method has ranked the ontologies in the dataset in increasing order on their level or degree of complexity.

Table 5.6 Complexity Metrics for the Middle 10 Ontologies in the WSM Ranking

	Ranges of Ontology Complexity Metrics			
	$4 \leq$ DIP ≤ 24	**$1 \leq$ ANP ≤ 133**	**$1 \leq$ APL ≤ 6**	**$1 \leq$ TIP ≤ 58741**
O_{51}	5	9	2	248
O_{9}	11	9	2	610
O_{84}	4	37	1	249
O_{90}	11	10	2	47
O_{35}	13	11	1	1368

O_7	18	31	2	899
O_{76}	13	9	1	1340
O_{42}	12	12	2	359
O_{75}	12	9	2	332
O_{60}	19	19	5	635

The WPM scores are provided in Table 5.7. The WPM scores in Table 5.7 resulted in the WPM ranking of ontologies in the dataset as in Figure 5.14.

Table 5.7 WPM scores

Index	Score	Index	Score	Index	Score
O_1	0.2558262002794407	O_{33}	0.2857058192415867	O_{70}	0.18139891550339649
O_2	0.20296976642072245	O_{34}	0.19393796547062972	O_{71}	0.4982661583725982
O_3	0.20243411611094866	O_{35}	0.24049985376143787	O_{72}	0.1957458829256612
O_4	0.5500340948279197	O_{36}	0.31672158445409887	O_{73}	0.38456859200267174
O_5	0.19881499503976463	O_{40}	0.1899071862114609	O_{75}	0.20079320013816004
O_7	0.22778364457756	O_{42}	0.21806235895723566	O_{76}	0.2426794500784851
O_8	0.3319336610346457	O_{43}	0.2054438295861994	O_{78}	0.22685510518861374
O_9	0.2060707525099263	O_{44}	0.16134695488727263	O_{79}	0.4934844898915937
O_{11}	0.535494388091431	O_{45}	0.1929922306789902	O_{80}	0.19789213811700018
O_{12}	0.23057069933145644	O_{46}	0.40080028545084323	O_{82}	0.33151762603927526
O_{13}	0.18910235632630673	O_{48}	0.207263533658338	O_{84}	0.22450197669459496
O_{14}	0.49700719112789143	O_{49}	0.26177531557329903	O_{85}	0.5193252272927013
O_{16}	0.5063501711837455	O_{50}	0.24564438469120775	O_{86}	0.20352093064950924
O_{17}	0.21106599162064069	O_{53}	0.18819859811743342	O_{68}	0.18604643981686422
O_{18}	0.2719355033894561	O_{54}	0.17640069864838617	O_{89}	0.19933844739431628
O_{20}	0.2723697277419254	O_{55}	0.30040482997058804	O_{90}	0.20325172680919676
O_{21}	0.26458133537906764	O_{56}	0.1937287152039094	O_{91}	0.21065000269720105
O_{24}	0.513913507491655	O_{57}	0.28374603664051634	O_{94}	0.18703583608459468
O_{26}	0.32526226722167784	O_{60}	0.4195485366074861	O_{95}	0.28756150498087346
O_{28}	0.19129082385995413	O_{61}	0.21810409049330423	O_{96}	0.16686241000959148
O_{29}	0.5414813227531798	O_{63}	0.3899244550447585	O_{97}	0.1834411623351328
O_{30}	0.19577998532901225	O_{66}	0.17958974730136087	O_{98}	0.2539747233733483
O_{32}	0.19566260862991208	O_{69}	0.15845070589800092	O_{99}	0.20701942587746378
				O_{100}	0.3219235011156542

The ranking results in WPM are similar to that of WSM in Figure 5.13. In fact, the majority of ontologies in the first and middle 10 positions and all the ontologies in the last 10 positions in the WSP ranking (Figure 5.13) are the same in the WPM ranking (Figure 5.14) despite slight differences in their positions.

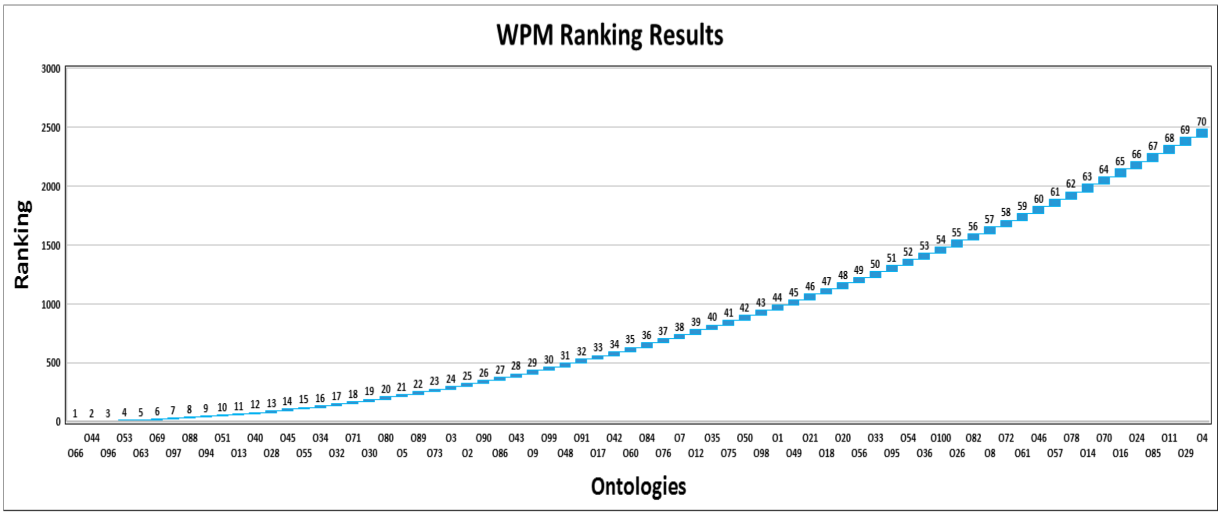

Figure 5.14 WPM Ranking Results

5.6.2 TOPSIS

The weighted normalised decision matrix for the TOPSIS method is presented in Appendix C. The scores obtained by applying the TOPSIS method on the normalised data are given in Table 5.8 and the resulting ranking results in Figure 5.15.

Table 5.8 TOPSIS Scores

Index	Score	Index	Score	Index	Score
O_1	0.21464580220511226	O_{33}	0.2841359053837396	O_{70}	0.12769324259930143
O_2	0.15864679810801133	O_{34}	0.0823384758807162	O_{71}	0.5994893949255741
O_3	0.12176670738458344	O_{35}	0.18359246669492574	O_{72}	0.1631442795528729
O_4	0.5723862763984412	O_{36}	0.3008222375691712	O_{73}	0.4455530250085954
O_5	0.09906522688706627	O_{40}	0.14407867763417898	O_{75}	0.16855002300329125
O_7	0.2757300713420944	O_{42}	0.18749700659372145	O_{76}	0.1884778801247229
O_8	0.3725362235057307	O_{43}	0.08023481920952984	O_{78}	0.22279969789607015
O_9	0.16709630177198379	O_{44}	0.028267527200177014	O_{79}	0.5110257438046172
O_{11}	0.4804465493812897	O_{45}	0.14851502382993798	O_{80}	0.11416567997491807
O_{12}	0.3305076192346145	O_{46}	0.45568775755374635	O_{82}	0.36646829026018696
O_{13}	0.10739180907175269	O_{48}	0.09345210369309348	O_{84}	0.19758089650394842
O_{14}	0.5135917105219867	O_{49}	0.2258201748325731	O_{85}	0.502847409834693
O_{16}	0.49247561904736	O_{50}	0.20447058380734537	O_{86}	0.1474091633906704
O_{17}	0.13935474055863417	O_{53}	0.1567721121537718	O_{68}	0.08017412844085793
O_{18}	0.19872427935103304	O_{54}	0.10441973944860099	O_{89}	0.13525688620927265
O_{20}	0.2513688318449301	O_{55}	0.2953426772722449	O_{90}	0.17185443308183146
O_{21}	0.25230841088109784	O_{56}	0.1208881996742813	O_{91}	0.09442830661785806
O_{24}	0.4737599902176345	O_{57}	0.2646753777502893	O_{94}	0.09136671639665392
O_{26}	0.2934217980525884	O_{60}	0.4115406726178889	O_{95}	0.2803821255766595
O_{28}	0.1167860950954815	O_{61}	0.19644446851731892	O_{96}	0.08251737089280844
O_{29}	0.5682783145096328	O_{63}	0.34978875092450784	O_{97}	0.046595386537718636

O_{30}	0.09782240615839766	O_{66}	0.11802763382845911	O_{98}	0.20388158213274304
O_{32}	0.15474765141893906	O_{69}	0.06510585505696595	O_{99}	0.10118532823075109
				O_{100}	0.3893317335071018

The ranking results of TOPSIS are similar to that of WSM and WPM methods in that, the majority of ontologies in the first and middle 10 positions and all the ontologies in the last 10 positions in the TOPSIS ranking (Figure 5.13) are the same in the WSM and WPM rankings (Figures 5.13 and 5.12) despite slight differences in their positions.

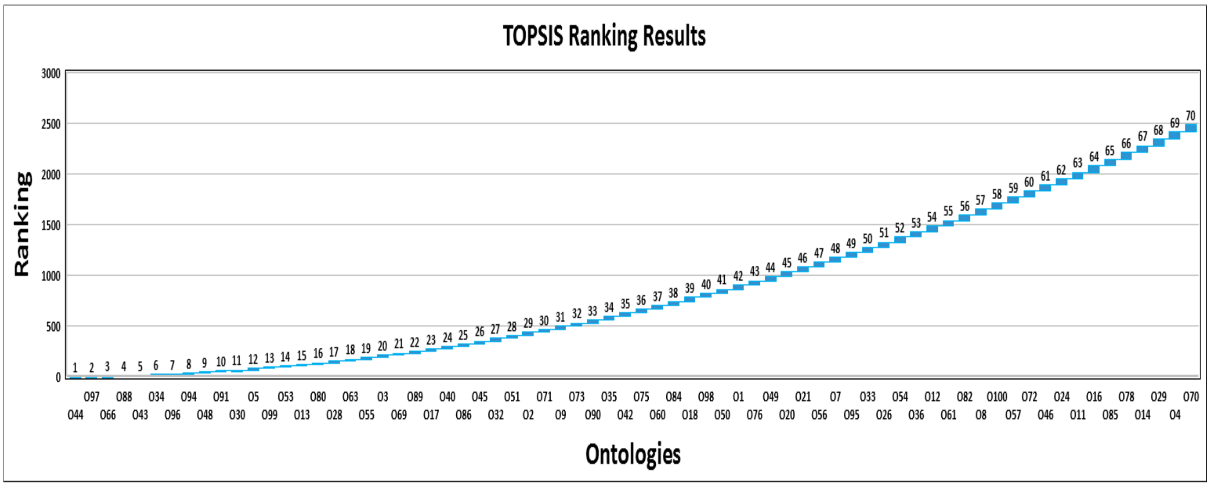

Figure 5.15 TOPSIS Ranking Results

5.6.3 WLCRT

The normalised decision matrix for the WLCRT method is presented in Appendix B. This matrix was used to compute the Pearson correlation coefficients and resulted in the proximity matrix in Table 5.9.

Table 5.9 Proximity Matrix of the Complexity Metrics

	DIT	ANP	APL	TIP	EOG	RR	CR	SOV
DIT	1.0	0.5477	0.6797	0.0307	0.5192	-0.0369	-0.2086	0.1531
ANP	0.5477	1.0	0.6297	0.1774	0.2395	-0.1245	-0.1252	0.1535
APL	0.6797	0.6297	1.0	0.101	0.4263	-0.3046	-0.1011	0.0361
TIP	0.0307	0.1774	0.101	1.0	0.2313	-0.2967	-0.1203	0.7917
EOG	0.5192	0.2395	0.4263	0.2313	1.0	-0.023	-0.0512	0.1568
RR	-0.0369	-0.1245	-0.3046	-0.2967	-0.023	1.0	0.1286	0.0351
CR	-0.2086	-0.1252	-0.1011	-0.1203	-0.0512	0.1286	1.0	-0.069
SOV	0.1531	0.1535	0.0361	0.7917	0.1568	0.0351	-0.069	1.0

Thereafter, the proximity matrix in Table 5.9 was used to computes the eigenvalues and their corresponding eigenvectors as specified in the Equation 3.25 of Chapter 3, Subsection 3.4.5.2. The resulting eigenvalues and eigenvectors were further used to compute the WLCRT scores as in Table 5.10.

Table 5.10: WLCRT Scores

Index	**Score**	**Index**	**Score**	**Index**	**Score**
O_1	0.7751382435733748	O_{33}	0.647571782868717	O_{70}	0.6475717778983738
O_2	0.733866739972547	O_{34}	0.6475717778983621	O_{71}	0.8107307525997974
O_3	0.7651330305791356	O_{35}	0.7138563151599854	O_{72}	0.7226108754545635
O_4	0.8118525404552035	O_{36}	0.7688849854530634	O_{73}	0.7376186949014842
O_5	0.8008141214193749	O_{40}	0.6475717779019686	O_{75}	0.7376186948453375
O_7	0.6913445847475866	O_{42}	0.7113557421497174	O_{76}	0.7288660851950229
O_8	0.8114071406921174	O_{43}	0.7838928049430887	O_{78}	0.6800887201292144
O_9	0.6600804997517581	O_{44}	0.6475717778983621	O_{79}	0.8118525438069482
O_{11}	0.733131479723002	O_{45}	0.6825900233817765	O_{80}	0.806404534179833
O_{12}	0.8088441578994083	O_{46}	0.8116421812828734	O_{82}	0.7776395468243947
O_{13}	0.6825900233777413	O_{48}	0.6475717778983767	O_{84}	0.8326682182893675
O_{14}	0.8118525438069482	O_{49}	0.7088575857604681	O_{85}	0.80157980568952
O_{16}	0.7938980443985821	O_{50}	0.7101043591115395	O_{86}	0.8426806097170503
O_{17}	0.796399321185729	O_{53}	0.7476249478320522	O_{68}	0.6475717778983621
O_{18}	0.6479002950381963	O_{54}	0.7238615269784391	O_{89}	0.7676343338314277
O_{20}	0.8489266894047938	O_{55}	0.6700835072761373	O_{90}	0.6900982405812793
O_{21}	0.7939069805099886	O_{56}	0.7463732562151842	O_{91}	0.6938458879961137
O_{24}	0.733131479723002	O_{57}	0.8090061243676567	O_{94}	0.6475717778983643
O_{26}	0.7526265143365002	O_{60}	0.7689014754513359	O_{95}	0.7201095721056539
O_{28}	0.7651330305791356	O_{61}	0.7288660851950229	O_{96}	0.6475717778983621
O_{29}	0.8118525404552035	O_{63}	0.7688981403565219	O_{97}	0.6475717778983621
O_{30}	0.8008141214193749	O_{66}	0.7838928049431436	O_{98}	0.6589415248251693
O_{32}	0.6475717779059186	O_{69}	0.6888432814990595	O_{99}	0.8214123536709959
				O_{100}	0.788895411440143

Figure 5.16 shows that the ranking results for the WLCRT method is slightly different from that of WSM, WPM and TOPSIS. In fact, 6 ontologies out of 10 in the first 10 positions of the WLCRT ranking in Figure 5.16 are the same in the WSP, WPM and TOPSIS ranking results (Figures 5.13, 5.14 and 5.15). Only 2 ontologies out of 10 in the middle 10 positions and 5 ontologies out of 10 in the last 10 positions of the WLCRT ranking appear in the ranking results of WSP, WPM and TOPSIS methods.

Table 5.11 Complexity Metrics of 4 Ontologies in the First 10 Positions of WLCRT Ranking Not Part of WSM, WPM and TOPSIS Rankings

	Ranges of Ontology Complexity Metrics			
	4 ≤ DIP ≤ 24	**1 ≤ ANP ≤ 133**	**1 ≤ APL ≤ 6**	**1 ≤ TIP ≤ 58741**
O_{34}	6	1	1	5
O_{48}	5	3	1	1
O_{40}	10	9	2	15
O_{32}	11	8	2	83

However, the analysis of the complexity metrics (Appendix A) of the 4 ontologies in the first 10 positions of the WLCRT method (O_{32}, O_{34}, O_{40}, O_{48}) that do not appear in the ranking results of WSM, WPM and TOPSIS reveals that they have smaller values for DIT, ANP, APL and TIP as in Table 5.11; this indicates that they are less complex like the other ontologies in the first 10 positions in the ranking results of WSM, WPM and TOPSIS methods. This finding is supported in Figure 5.1 and 5.2 where these ontologies hold a low number of classes and properties.

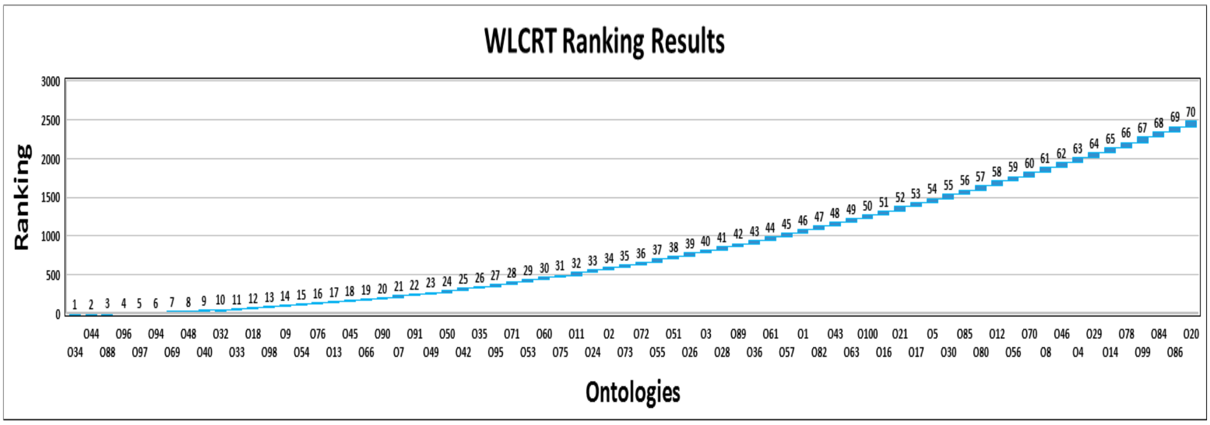

Figure 5.16 WLCRT Ranking Results

Similarly, the analysis of the complexity metrics (Appendix A) of the 8 ontologies (O_{11}, O_{24}, O_{2}, O_{73}, O_{72}, O_{55}, O_{26}, O_{3}) in the middle 10 positions and 5 ontologies (O_{8}, O_{20}, O_{84}, O_{86}, O_{99}) in the last 10 positions in the WLCRT ranking that do not appear in the ranking results of the WSM, WPM and TOPSIS methods. This reveals that they have relatively high values for the DIT, ANP, APL and TIP (Table 5.12) compared to the ontologies in the first 10 positions (See part in Table 5.11); this is an indication that they are complex ontologies. This finding is further evidenced in Figures. 5.1 and 5.2 where these ontologies have a higher number of classes and properties.

Table 5.12 Complexity Metrics of 8 Ontologies in the Middle 10 Positions and 5 Ontologies in Last 10 Positions of WLCRT Ranking Not Part of WSM, WPM and TOPSIS Rankings

	Ranges of Ontology Complexity Metrics			
	4 ≤ DIP ≤ 24	**1 ≤ ANP ≤ 133**	**1 ≤ APL ≤ 6**	**1 ≤ TIP ≤ 58741**
	Middle 10 Positions of WLCRT			
O_{11}	19	44	5	2919
O_{24}	19	45	5	5176
O_2	11	10	2	813
O_{73}	14	86	2	110
O_{72}	11	6	2	121
O_{55}	14	7	4	210
O_{26}	16	29	3	2418
O_3	9	5	1	5124
	Last 10 Positions of WLCRT			
O_8	24	14	2	12414
O_{20}	21	75	6	7664
O_{84}	4	37	1	249
O_{86}	7	2	1	191
O_{99}	6	3	1	257

5.6.4 ELECTRE

The first step in the application of the ELECTRE method on the dataset consisted in computing the concordance and discordance matrices (Appendix D). These matrices were further used to compute the ELECTRE scores and ranking results in Table 5.13 and Figure 5.17, respectively.

Table 5.13 ELECTRE Scores

Index	Score	Index	Score	Index	Score
O_1	0.64	O_{33}	0.67	O_{70}	0.6
O_2	0.55	O_{34}	0.48	O_{71}	0.68
O_3	0.49	O_{35}	0.6	O_{72}	0.61
O_4	0.67	O_{36}	0.61	O_{73}	0.68
O_5	0.43	O_{40}	0.62	O_{75}	0.62
O_7	0.51	O_{42}	0.65	O_{76}	0.6
O_8	0.6	O_{43}	0.42	O_{78}	0.61
O_9	0.57	O_{44}	0.58	O_{79}	0.64
O_{11}	0.65	O_{45}	0.6	O_{80}	0.54
O_{12}	0.69	O_{46}	0.61	O_{82}	0.62
O_{13}	0.5	O_{48}	0.54	O_{84}	0.64
O_{14}	0.65	O_{49}	0.65	O_{85}	0.66
O_{16}	0.68	O_{50}	0.62	O_{86}	0.6
O_{17}	0.49	O_{53}	0.62	O_{68}	0.56
O_{18}	0.58	O_{54}	0.61	O_{89}	0.54
O_{20}	0.58	O_{55}	0.66	O_{90}	0.61
O_{21}	0.6	O_{56}	0.5	O_{91}	0.55
O_{24}	0.67	O_{57}	0.58	O_{94}	0.61
O_{26}	0.58	O_{60}	0.67	O_{95}	0.63
O_{28}	0.48	O_{61}	0.62	O_{96}	0.66
O_{29}	0.67	O_{63}	0.65	O_{97}	0.48
O_{30}	0.42	O_{66}	0.64	O_{98}	0.63
O_{32}	0.61	O_{69}	0.62	O_{99}	0.54
				O_{100}	0.66

The same findings as that obtained for the WLCRT method can be derived for the ELECTRE method. In fact, 6 ontologies out of 10 (O_3, O_5, O_{13}, O_{17}, O_{28}, O_{55}) in the first 10 positions of the ELECTRE method in Figure 5.17 do not form part of the first 10 positions of the ranking results for WLCRT, WSM, WPM and TOPSIS. However, these 6 ontologies can be classified as less complex because they have smaller values for DIT, ANP, APL and TIP (Appendix A) as well as low number of classes and properties in Figure 5.1 and 5.2.

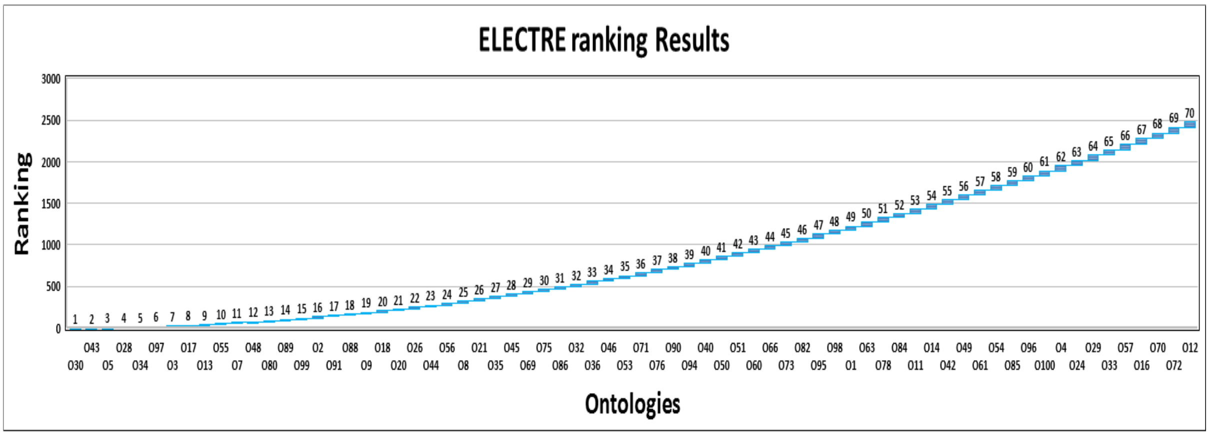

Figure 5.17 ELECTRE Scores

Similarly, 8 ontologies (O_{86}, O_{32}, O_{36}, O_{46}, O_{53}, O_{71}, O_{94}, O_{40}) out of 10 in the middle positions and 4 ontologies out of 10 (O_{12}, O_{33}, O_{72}, O_{100}) in the last 10 positions of the ELECTRE ranking results (Figure 5.17) are not part of the ranking results of WLCRT, WSM, WPM and TOPSIS, in the middle and last 10 positions, respectively. However, the analysis of the complexity metrics (Appendix A) of these ontologies portrays that they have relatively higher values for DIT, ANP, APL and TIP (Table 5.14) and hold large number of classes and properties in Figure 5.1 and 5.2. Therefore, these ontologies are complex than those in the first 10 positions of the ELECTRE ranking. In particular, the ontology in the dataset with the highest number of classes (O_{12} in Figure 5.1) is among the 4 ontologies in the last 10 positions of ELECTRE ranking (Figure 5.17) that are missing in the last 10 positions of the WLCRT, WSM, WPM and TOPSIS ranking.

Table 5.14 Complexity Metrics of 8 Ontologies in the Middle 10 Positions and 4 Ontologies in Last 10 Positions of ELECTRE Ranking Not Part of WSM, WPM, TOPSIS and WLRCT Rankings

	Ranges of Ontology Complexity Metrics			
	$4 \leq$ DIP ≤ 24	**$1 \leq$ ANP ≤ 133**	**$1 \leq$ APL ≤ 6**	**$1 \leq$ TIP ≤ 58741**
	Middle 10 Positions of WLCRT			
O_{86}	7	2	1	191
O_{32}	11	8	2	83

O_{36}	19	10	3	1363
O_{46}	35	27	3	569
O_{53}	9	6	2	234
O_{71}	21	133	4	307
O_{94}	7	3	1	98
O_{40}	10	9	2	15
	Last 10 Positions of WLCRT			
O_{8}	24	14	2	12414
O_{20}	16	18	2	7664
O_{84}	4	37	1	249
O_{86}	7	2	1	191

In light of the discussions of the ranking results of the 5 MADM methods (WSM, WPM, TOPSIS, WLCRT, ELECTRE) above, one can conclude that these methods all ranked the ontologies in the dataset in ascending order on their level or degree of complexity.

5.7 VALIDATION OF RESULTS

The analysis of the advanced complexity metrics of biomedical ontologies in the dataset in Subsection 5.4.2, portrayed that the majority of these ontologies have a large size of vocabulary (SOV), and bigger average path length (APL) and entropy of ontology graph (EOG). These findings indicate that the biomedical ontologies in the dataset are highly complex (Yang et al. 2006, Zhang et al. 2010). Furthermore, the ontologies in the dataset were successfully ranked in Section 5.6 in increasing order on the aggregation of their complexity metrics by 5 MADM methods. The ranking results constitute important guidelines for the selection and reuse of biomedical ontologies in the dataset. It would therefore be advised to consider the reuse and sharing of these ontologies in the biomedical domain rather than trying to build similar ontologies *de novo*; the reuse may consist in using (1) parts of existing biomedical ontologies to build new ones or (2) the full ontologies in new applications (Ding et al. 2007). In fact, ontology reuse (1) reduces human efforts required to formalize new ontologies *de novo*, (2) increases the quality of the resulting ontologies because the reused ontologies have already been tested, (3) simplifies the mapping between ontologies built using shared components of existing ontologies, and (4) improves the efficiency of ontology maintenance (Ding et al. 2007). Furthermore, the analysis of the tree impurity (TIP), relationship richness (RR) and class richness (CR) metrics in Subsection 5.4.2 revealed that the biomedical ontologies in the dataset can be easily reused and maintained (Tartir et al. 2005; Zhang et al. 2010; Sugumaran and Gula 2012). These findings are supported by the fact that the biomedical ontologies concerned are available for download free of charge on the BioPortal repository and many researchers (Salvadores 2013; Whetzel and Team 2013), including this study, provide metadata that may be useful in understanding, reusing, sharing and maintaining these ontologies in the biomedical domain.

5.8 CONCLUSION

This chapter presented the experimental results of the application of the framework for analysing the complexity of ontologies drawn in Chapter 4 Section 4.2. The Chapter begins with the collection of 100 biomedical ontologies from the BioPortal repository. Thereafter, the primitive and advanced complexity metrics of the collected ontologies were computed and analysed. The advanced complexity metrics were further used to rank the ontologies with 5 MADM methods. Finally, the impact of the advanced complexity metrics and the ranking results was analysed to validate the proposed framework. The next chapter concludes the study and provides directions for future research.

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 SUMMARY OF THE STUDY

The aim of this study was to design and apply a framework for analysing the complexity of ontologies. The framework was designed and specified in Chapter 4, Section 4.2. It has 4 phases or stages, namely, ontology acquisition, complexity metrics computation, ontology ranking and validation. The first phase of the framework was achieved through the download of 100 biomedical ontologies from the BioPortal repository to constitute the dataset for the study as in Chapter 5, Section 5.2. The second phase of the framework, namely, complexity metrics computation, consisted in carrying out in Chapter 2, a comprehensive detailed review of the ontology evaluation discipline with a focus on existing complexity metrics for measuring the complexity of ontologies. This review of the *state-of-the-art* of ontology complexity analysis revealed that implementing computer programmes for the empirical analysis of ontology complexity metrics through the automatic exploration of RDF or OWL ontology graphs remained challenging.

To solve the abovementioned problem, nine generic algorithms for computing ontology complexity metrics through the processing of RDF graphs were designed in Chapter 3, Subsection 3.3.3 to compute ontology complexity metrics including the depth of inheritance (DIP), size of the vocabulary (SOV), entropy of ontology graphs (EOG), the average part length (APL) and average number of paths per class (ANP), the tree impurity (TIP), relationship richness (RR) and class richness (CR). These algorithms were implemented to calculate the complexity metrics of biomedical ontologies in the dataset as in Appendix A. The resulting complexity metrics permitted the analysis of the level or degree of complexity of the biomedical ontologies in the dataset (Chapter 5, Subsection 5.4.2). The execution times of the proposed algorithms were further analysed to measure their performances in Chapter 5, Section 5.5.

To provide guidelines for the selection and reuse of ontologies in the dataset based on the aggregation of their complexity metrics, in the ontology ranking phase (3rd phase) of the proposed framework in Chapter 4, Section 4.2, 5 Multi-attributes Decision Making (MADM) methods, namely, Weighted Sum Method (WSM), Weighted Product Method (WPM), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), Weighted Linear Combination Ranking Technique (WLCRT) and Elimination and Choice Translating Reality (ELECTRE) were applied to rank the Biomedical ontologies in the dataset. The results showed that the 5 MADM methods (WSM, WPM, TOPSIS, WLCRT, and ELECTRE) successfully ranked the ontologies in the dataset in ascending order on their levels or degrees of complexity (Chapter 5, Section 5.6).

The last phase of the framework, namely, validation was achieved in Chapter 5, Section 5.7 by drawing the summary of the results of the previous phases of the framework and their impact on the issues of selection and reuse of the Biomedical ontologies in the dataset.

6.2 LIMITATIONS, RECOMMENDATIONS AND FUTURE WORK

The framework for the analysis of complexity of ontology developed in this study enables computation of metadata (Complexity metrics) for assessing the level or degree of complexity of ontologies in a domain of knowledge. However, the proposed framework presents some limitations:

- **Ontology Languages**: The algorithms designed for the computation of the ontology complexity metrics can process only ontologies written in RDF or OWL ontology languages. Although most ontologies are developed using these two languages, there is a considerable number of ontologies available in the public domain written using other languages such the OBO, DAML+OIL, SHOE, etc (Salvadores et al. 2011). Future research can be carried out to help make the proposed algorithms implementable using any ontology language
- **Ontology Search:** To analyse the complexity of ontologies using the proposed framework, users must collect the content of ontologies on the Internet and manually input them in the developed algorithms. A future direction of research can look at directly connecting Semantic Web search engines to the programme that implements the algorithms to enable an automatic search and input of ontologies into the system.
- **Ontology Exploration**: the proposed algorithms for computing the complexity metrics of ontologies were implemented using the Java Jena API. Future research could focus on implementing the algorithms in other Semantic Web APIs such as Sesame and Allegrograph (Ramanujam et al. 2009; Fan et al. 2010; Zhou 2010).

6.3 CONCLUSION

This study presented a framework for the analysis of complexity of ontologies. The framework was applied on a dataset of 100 biomedical ontologies collected from the BioPortal repository. The application of the framework implemented a set of algorithms to compute the complexity metrics of ontologies in the dataset. The performance evaluation results of these algorithms were submitted for review to *International Journal of Semantic Web and Information Systems (IJSWIS)*. The analysis of the complexity metrics of the biomedical ontologies in the dataset, portrayed that the majority of these ontologies have a large size of vocabulary (SOV), and bigger average path length (APL) and entropy of ontology graph (EOG). These findings indicate that the biomedical ontologies in the dataset are highly complex. These results and findings were published in Kazadi & Fonou-Dombeu (2016a). Furthermore, the ontologies in the dataset were successfully ranked in Section 5.6 in increasing order on the aggregation of their complexity metrics by 5 MADM methods; this work was published in Kazadi & Fonou-Dombeu (2016b). The ranking results constitute important guidelines for the selection and reuse of biomedical ontologies in the dataset. Although the proposed framework in this study has been applied in the biomedical domain, it could be applied in any other domain of Semantic Web to analyze the complexity of ontologies.

REFERENCES

ALANI, H. & BREWSTER, C. (2006) “Metrics for Ranking Ontologies”, In the Proceedings of the 4th International Conference on Evaluation of Ontologies for the Web, Edinburgh, Scotland, 22 May , pp 34-40.

ANAND N., VAN DUIN J., TAVASSZY L.& WIGAN M. (2014) “About quality of semantic data for the city logistics domain: A comparison with the stakeholders’ perspectives”, In Proceedings of the Transportation Research Board 93rd Annual Meeting, Washington D.C, USA, 12-16 January, pp 15-26.

ARORA, S AND BARAK, B. (2009) Computational complexity: a modern approach, *1st Edition, Cambridge University Press.*

BENAYOUN, R., ROY, B. & SUSSMAN, N. (1966) “Manual de reference du programme electre, Note de Synthese et Formation, No. 25 ”, *Direction Scientifique SEMA*, Paris, France.

BEN ABBES, S. (2009) “Evaluation d’ontology”, Master Degree thesis, Laboratory of Computer Science of Paris North, Department of Intelligent systems.

BONDI, A.B. (2000) “Characteristics of scalability and their impact on performance” In Proceedings of the second International Workshop on Software and Performance, Ottawa,Canada, 17-20 September, pp. 55-63.

BONTAS, E. P., MOCHOL, M., & TOLKSDORF, R. (2006) “Case study on ontology reuse” In Proceeding of the International Conference on Knowledge Technologies and Data-driven Business (I-KNOW), Graz, Greece, 29 June- 1 July, pp. 153-160.

BORGO, S. (2006) “How formal ontology can help civil engineers, Studies in Computational” *Intelligence, in: Teller, J., Lee, J., Roussey, C. (eds.) Ontologies for Urban Development: Interfacing Urban Information Systems. Studies in Computational Intelligence*, vol. 61, pp. 143–156. University of Geneva 6,7 Nov 2006.

BRANK, J., GROBELNIK M. & MLADENIC, D. (2005) "A survey of ontology evaluation techniques" In Proceedings of the Conference on Data Mining and Data Warehouses, Germany, Berlin, 22-26 August pp. 345-350.

BREWSTER, C., ALANI, H., DASMAHAPATRA, S. & WILKS, Y. (2004) "Data driven ontology evaluation", In Proceedings of International Conference on Language Resources and Evaluation (LREC 2004), Lisbon, Portugal, 26-28 May, pp. 31-44.

CARROLL, J., DICKINSON, I., DOLLIN, C., REYNOLDS, D., SEABONE, A. & WILKINSON, K. (2004) "Jena: implementing the semantic web recommendations" In Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, New York, USA, 17-22 May, pp. 74-83.

CHOU, J.R. (2013) "A weighted linear combination ranking technique for multi-criteria decision analysis", *South African Journal of Economic and Management Sciences* Special Issue 16, pp. 28-41.

COLLINS, J. B. & CLARK, D. (2004) "Towards an Ontology of Physics", In Proceedings of the European Simulation Interoperability Workshop, Edinburgh, Scotland, 28 June- 1 July, pp. 55-63.

CORCHO, O. & GOMEZ-PEREZ, A. (2000) "A road map to ontology specification languages", In Proceedings of the 12th International Conference on Knowledge Acquisition, Modeling and Management, Juan-les-Pins ,France,2-6 October pp. 80-96.

CROSS, V. & PAL, A (2008) "An ontology analysis tool", International Journal of General Systems, Vol. 37, No. 1, pp. 17-44.

DAMERON, O., BETTEMBOURG, C. & LE MEUR, N. (2013) "Measuring the Evolution of Ontology Complexity: The Gene Ontology Case Study", *PLoS ONE,* Vol. 8, No. 10, pp 92-105.

DING, L., FININ, T., JOSHI, A., PAN, R., COST, R.S., SACHS, J., PENG, Y. & REDDIVARI, P. (2005) "Swoogle: a search and metadata engine for the semantic web", In

Proceedings of the 13th ACM Workshoperence on Information and Knowledge Management, New York ,USA,31 October- 05 November, pp.652-659.

DUQUE-RAMOS, A., FERNÁNDEZ-BREIS, J.T., INIESTA, M., DUMONTIER, M., ARANGUREN, E., SCHULZ, S., AUSSENAC-GILLES, N., & STEVENS, R. (2013) “Evaluation of the OQuaRE framework for ontology quality”, *Expert Systems with Applications*, Vol. 40, No.7, pp. 2696–2703.

D'AQUIN, M. & MOTTA, E. (2011) “Watson, more than a Semantic Web search engine”, *Semantic Web Journal*, Vol. 2, No. 1, pp. 55-63.

ERMATITA, Z., HARTATI, S., WARDOYO, R. & HARJOKO, A. (2012) “ELECTRE-Entropy method in Group Decision Support System Modelto Gene Mutation Detection” *International Journal of Advanced Research in Artificial Intelligence (IJARAI)*, Vol. 1, No. 1, pp.58-63.

ESPOSITO, A., ZAPPATORE, M. & TARRICONE, L. (2011) “Evaluating scientific domain ontologies for the electromagnetic knowledge domain: a general methodology”, *International Journal of Web & Semantic technology (IJWesT)*, Vol. 2, No. 3, pp. 1-19.

FAN, X., ZHANG, P. & ZHAO, J. (2010) “Transformation of Relational Database Schema to Semantics Web Model”, In Proceedings of the Second International Conference on Communication Systems, Networks and Applications, Hong Kong, China, 29 June- July 1,pp. 379-384.

FISHBURN, P.C. (1967) “Additive Utilities with Incomplete Product Set: Applications to Priorities and Assignments”, *Operations Research Society of America (ORSA) Publication*, Baltimore, MD.

GANGEMI, A., CATENACCI, C., CIARAMITA, M. & LEHMANN, J. (2005) “A theoretical framework for ontology evaluation and validation”, In proceedings of Semantic Web Access and Personalisation (SWAP), Trento,Italy, 9 September, pp. 156-170.

GROZA, A., DRAGOSTE, I., SINCAI, I. & JIMBOREAN, I. (2014) "An Ontology Selection and Ranking System Based on the Analytic Hierarchy Process", In Proceedings of the Intelligent Computer Communication and Processing (ICCP), Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Cluj, Romania, 4-6 September, pp 293-300.

GRUBER, T.R. (1993) "Toward principles for the design of ontologies used for knowledge sharing", *International Journal of Human-Computer Studies*, Vol. 43, No. 1, pp. 907-928.

GUARINO, N., & BOLDRIN, L. (1993) "Ontological Requirements for Knowledge Sharing", In Proceedings of the IJCAI93 Workshop on Knowledge Sharing and Information Interchange, Chambery, France, 22- 24 August, pp. 31-46.

GUIZZARDI, G., FALBO, R. A., & PEREIRA, F.J.G. (2002) "Using objects and patterns to implement domain ontologies", *Journal of Brazilian Computer Society*, Vol.8, No. 1, pp.43-56.

HAUKE, J. & KOSSOWSKI, T. (2011) "Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data", *Quaestiones geographicae* Vol. 30, No. 2, pp.87-93.

ISO25000 (2005): ISO/IEC 25000 2005, Software engineering – Software product quality requirements and evaluation (SQuaRE) – guide to square (ISO/IEC 25000). Geneva, Switzerland: International Organization for Standardization.

JIANG, J., TANG, D.W. & CHEN, Y.W (2010) "Topsis with belief structure for group belief multiple criteria decision making", *International Journal of Automation and Computing*, Vol. 7, No. 3, pp. 359-364.

KANG D., XU B., LU J. & CHU W.C. (2004) "A Complexity Measure for Ontology Based on UML" In Proceedings of 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04), Suzhou, China,26-28 May, pp. 222-228.

KALIBATIENE, D., VASILECAS, O. & GUIZZARDI, G. (2007) “Transforming Ontology Axioms to Information Processing Rules – An MDA Based Approach”, *Advances in Information Systems Development 2*007, Springer, New York, pp 493-500.

KAZADI, Y.K. & FONOU-DOMBEU, J.V. (2016a) Analysis of Advanced Complexity Metrics of Biomedical Ontologies in the BioPortal Repository, *In Proceedings of the 1st International Conference on Complex Information Systems (COMPLEXIS 2016)*, Rome, Italy, April, pp. 107-104.

KAZADI, Y.K. & FONOU-DOMBEU, J.V. (2016b) Complexity Based Ranking of Biomedical Ontologies, *In Proceedings of the 3rd IEEE International Conference on Advances in Computing, Communication & Engineering 2016 (ICACCE 2016),* Durban, South Africa, ISBN: 987-1-5090-2576-6, 28-29 November, pp. 423-429.

LASSILA, O. & MCGUINESS, D. (2001) “The Role of Frame-Based Representation on the Semantic Web”, *Stanford Knowledge Systems Laboratory Technical Report KSL-01-02*.

LETHBRIDGE, T. (1998) “Metrics for Concept-Oriented Knowledge Bases”, *International Journal of Software Engineering and Knowledge Engineering*, Vol. 8, No.2, pp.161-188.

LINDBERG, D.A.B., H UMPHREYS, B.L. & RAY, A.T. (1993), “The unified medical language system”, *Methods of Information in Medicine,* Vol. 32, No. 1, pp. 281 – 291.

LOZANO-TELLO, A., GOMEZ-PEREZ, A. & SOSA, E.(2004) “Selection of Ontologies for the Semantic Web”, InProceedings of the International Conference of Web Engineering (ICWE), Oviedo, Spain, 28-30 July, pp 88-91.

LU, Q. (2006) “A Support Tool for OWL Ontology Evaluation”, Master Degree Thesis, Concordia University, Department of Computer Science and Software Engineering.

MAEDCHE, A. & STAAB, S. (2002) “Measuring similarity between ontologies”, In Proceedings of the International Conference on Knowledge Engineering and Knowledge Management, Berlin, Germany, 1 October, pp 251-263.

MAIGA, G. & DDEMBE, W. (2009) “Flexible Biomedical Ontology Selection Tool”, *International Journal of Computing and ICT Research*, Special Issue 3, No. 1, pp. 53-66.

MANSO M. E., GENERO, M. & PIATTINI, M. (2003) “Non-redundant metrics for UML class diagram structural complexity”, In Proceedings of 15^{th} international conference on advanced information systems Engineering, Thessaloniki, Greece, 16-20 June, pp. 50-65.

MILLER, D.W. & STARR, M.K. (1969) “Executive Decisions and Operations Research”, *Prentice-Hall, Inc.*, Englewood Cliffs, NJ.

MOWSHOWITZ, A. & DEHMER, A. (2012) “Entropy and the complexity of graphs revisited”, *Entropy*, Vol. 14, No. 3, pp. 559–570.

NATALYA F. N. (2004) “Evaluation by Ontology Consumers”, *IEEE Journal of Intelligent Systems*, Vol. 8, No. 1, pp. 80-81.

NECHES R., FIKES R. E., FININ T., GRUBER T. R., SENATOR T.& SWARTOUT W. R. (1990), “Enabling technology for knowledge sharing”, *AI Magazine*, Vol. 12, No. 3, pp. 36-56.

OBRST, L., ASHPOLE, B., CEUSTERS, W., MANI, I. & SMITH, B. (2007) “Semantic web chapter. The evaluation of ontologies - Toward Improved Semantic Interoperability”, *Springer*, New York, pp. 139–158.

PAK, J. & ZHOU, L. (2011) “Exploring the Grand Challenges for the Next Generation E-Business”, *Lecture note in Business Information Processing,* Vol. 52, No. 1, pp. 10-18.

PAL, A. (2005) “An Ontology Analysis Tool for Consumer” , Master Degree Thesis, University of Miami, Department of Computer Science and System Analysis.

PATEL, C., SUPEKAR, K., LEE, Y. & E. PARK (2003) “Ontokhoj: A semantic web portal for ontology searching, ranking, and classification”, In Proceedings of the 5th ACM

Inernational Workshop on Web Information and Data Management, New Orleans, USA, 3-8 November, pp. 58–61.

PENTLAND, A., MOGHADDAM, B. & STARNER, T. (1993) "View-based and modular eigenspaces for face recognition", Technical Report 245, MIT Media Lab Vismod.

PERONI, S., SHOTTON, D. & VITALI, F. (2013) "Tools for the automatic generation of ontology documentation: A task-based evaluation", *International Journal on Semantic Web and Information Systems (IJSWIS)*, Vol. 9, No. 1, pp. 21-44.

PORZEL, R. & MALAKA, R. (2004) "A task-based approach for ontology evaluation", In Proceedings of the 16th European Confenrence on Artificial Intelligence (ECAI 2004), Valencia, Spain, 7 August, pp. 982-996.

RAMANUJAM, S., GUPTA, A., KHAN, L., SEIDA, S. & THURAISINGHAM, B. (2009) "R2D: A Bridge between the Semantic Web and Relational Visualization Tools", In Proceedings of the 3rd IEEE International Conference on Semantic Computing (ICSC), Berkeley, USA, 14-16 September, pp. 303-311.

RAY, T. & TRIANTAPHYLLOU, E. (1998), "Evaluation of Rankings With Regard to the Possible Number of Agreements and Conflicts", *European Journal of Operational Research*, Vol. 106, No. 1, pp. 129-136.

RECTOR, A.L., NOWLAN, W.A., KAY, S., GOBLE, C. A. & HOWKINS, T.J. (1993) "A framework for modelling the electronic medical record", *Methods of Information in Medicine*, Vol. 32, No. 1, pp.109 – 119.

RODRIGUEZ, D, SICILIA, M. & GARCA, E. (2012) "Empirical Findings on Ontology Metrics", *International Journal of Expert Systems with Applications*, Vol. 39, No.1, pp. 6706-6711.

ROUSSEY, C., PINET, F., KANG, M.A. & CORCHO, O. (2011) "An Introduction to Ontologies and Ontology Engineering, in Ontologies in Urban Development Projects

(Advanced Information and Knowledge Processing)", G. Falquet, C. Métral, and J. Teller, Editors., *SpringerVerlag London Limited*, London.

SAATY, T.L. (1977) "A scaling method for priorities in hierarchical structures", *Journal of Mathematical Psychology*, Vol. 15, No. 1, pp. 57-68.

SALVADORES, M., ALEXANDER, P.R., MUSEN, M.A. & NOY, N.F. (2013) "BioPortal as a Dataset of Linked Biomedical Ontologies and Terminologies in RDF", *Semant Web*, Vol. 4, No. 3, pp. 277-284.

SMILIKOVA, R., & WACHIOWAK, M.P (2002) "Aggregation Operator for Selection Problems", *Journal Fuzzy Sets and Systems Special issue: Soft decision analysis*, Vol. 131, No. 1, pp. 23-34.

STOJANOVIC, L. & MOTIK, B. (2002), "Ontology Evolution within Ontology Editors" In Proceedings of OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management, Siguenza, Spain, 1-4 October, pp. 53-62.

SUDEEPTHI, G., ANURADHA, G. & BABU, M. (2012) "A Survey on Semantic Web Search Engine", *International Journal of Computer Science Issues (IJCSI)*, Vol. 9, No. 2, pp.1694-0814.

SUPEKAR, K., PATEL, C. & LEE, Y. (2004) "Characterizing Quality of Knowledge on Semantic Web", In Proceedings of AAAI Florida AI Research Symposium (FLAIRS-2004), Miami Beach, USA, 17- 19 May, pp. 472-478.

TARTIR S., ARPINAR B., MOORE, M., SHETH, A. & ALEMAN-MEZA, B. (2005) "OntoQA: Metric-Based Ontology Quality Analysis" In Proceedings of IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources, Houston, USA, 15 July, pp. 45 -53.

TARTIR, S. & ARPINAR, B. (2007) "Ontology evaluation and Ranking", In Proceedings of the International Conference on Semantic Computing, Irvin, USA, 17-19 September, pp. 185-192.

TU, S.W., ERIKSSON, H., GENNARI, J.H., SHAHAR, Y. & MUSEN, M.A. (1995) "Ontology-based configuration of problem-solving methods and generation of knowledge acquisition tools: the application of PROT EGE-II to protocol-based decision support", *Artificial Intelligence in Medicine*, Vol. 7, No 1, pp. 257 – 289.

USCHOLD, M. & KING, M. (1995) "Towards a Methodology for Building Ontologies" In Proceedings of the IJCAI'95Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, 20-25 August, pp. 52-71.

UYSAL, H.T. & YAVUZ, T (2014) "Selection of Logistics Centre Location via ELECTRE Method: A Case Study in Turkey", *International Journal of Business and Social Science*, Vol. 5, No. 9, pp.55-71.

VAN HEIJST, G., SCHREIBER, A. & WIELINGA, B. (1996) "Using explicit ontologies in KBS development", *International Journal of Human-Computer Studies. Forthcoming*, Vol. 46, No. 2-3, pp. 183-292.

VELARDI, P., NAVIGLI, R., CUCHIARELLI, A. & NERI, F. (2005) "Evaluation of ontolearn, a methodology for automatic population of domain ontologies" *Ontology Learning from Text: Methods, Applications and Evaluation.* In P. Buitelaar, P. Cimiano, and B. Magnini, editors, IOS Press. City of publisher

VILLALON, P., SUAREZ-FIGUEROA, M. & GOMEZ-PEREZ, M.C (2010) "Reusing Ontology Design Patterns in a Context Ontology Network", In Proceedings of the Second Workshop on Ontology Patterns (WOP 2010) co-located at ISWC 2010, Shangai, China, 08 November, pp. 31-44.

VOMEL, C., VEPREK, R.G., WEBER, U. & ARBENZ, P. (2008) "Iterative Solution of Generalized Eigenvalue Problems from Optoelectronics with Trilinos", *Technical Report 598, Institute of Computational Science*, ETH Zurich, June 2008.

WANG, Z., XIA, S. & NIU, Q. (2013) "A Novel Ontology Analysis Tool", *International Journal Applied Mathematics & Information Sciences,* Vol. 8, No.1, pp. 255-261.

WALLACE, S.W. (2000) Decision Making Under Uncertainty: Is Sensitivity Analysis of Any Use?, *Operations Research,* Vol. 48, No 1, pp.20-25.

WHETZEL P.L & TEAM N. (2013) "NCBO Technology: Powering semantically aware applications", *Journal of Biomedical Semantics* Vol. 4, No. 1, pp.88-95.

WBIO, Welcome to the NCBO Bioportal, [Online], Available at: http://bioportal.bioontology.org, Last Accessed: 28 November 2015.

WEYUKER, E.J (1988) "Evaluating software complexity measures", *IEEE transactions on Software Engineering,* Vol. 14, No. 9, pp.1357-1365.

YANG Z., ZHANG D. & YE C. (2006), "Evaluation Metrics for Ontology Complexity and Evolution Analysis", In Proceedings of the IEEE International Conference on e-Business Engineering, Coventry, UK, 24-26 October, pp. 162–170.

YAO, H., ORME, A. M. & ETZKORN, L. (2005) "Cohesion Metrics for Ontology Design and Application", *Journal of Computer Science,* Vol.1, No. 1, pp.107-113.

YU, W., LI, Q., CHEN, J. & CAO, J. (2007) "Os_rank: analysis for ontology ranking"**,** In Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering, Istanbul, Turkey, 11-15 April, pp. 339-346.

ZHANG H., LI Y. F., AND TAN H. B. K. (2010) "Measuring design complexity of semantic web ontologies", *The Journal of Systems and Software*, Vol.83, No. 3, pp.803-814.

ZHOU, S. (2010) "Relational Databases Access based on RDF View", In Proceedings of the 1stInternational Conference on E-Business and E-Government, Guangzhou, China, 7-9 May, pp. 5486-5489.

APPENDIX A: Complexity Metrics of Biomedical Ontology from the Dataset

Index	Ontology Name	DIT	ANP	APL	TIP	EOR	RR	CR	SOV
O_1	Information Consent Ontology	11	7	3	466	2.042	0.602	0.0022	962
O_2	Alzheimer's Disease Ontology	11	10	2	813	1.815	0.569	0	2351
O_3	Bone dysplasia Ontology	9	5	1	5124	1.776	0.594	0.0004	6563
O_4	Cigarette Smoke Exposure Ontology	21	75	6	122	2.244	0.500	0	20279
O_5	Ontology of vaccine advert events	4	3	1	2890	1.738	0.685	0.0	3070
O_7	Dengue Fever Ontology	18	31	2	899	1.040	0.535	0	5952
O_8	Galen Ontology	24	14	2	12414	2.213	0.631	0.0	68613
O_9	Human Dermatological Ontology Disease	11	9	2	610	2.036	0.510	0.0	3602
O_{11}	Natural Products Ontology	19	44	5	2919	2.329	0.500	0.0556	31554
O_{12}	NCI Thesaurus	6	4	2	58741	2.135	0.521	0.0042	112377
O_{13}	Ontology of Adverse Events	9	5	1	409	1.609	0.528	0	3016
O_{14}	Ontology of drug neuropathy adverse event	22	49	6	1533	2.304	0.558	0.0004	2465
O_{16}	Uber Anatomy Ontology	16	71	3	30562	2.311	0.617	0.0	42382
O_{17}	Vaccine Ontology	9	4	1	6571	2.032	0.619	0.00009	10706
O_{18}	Experimental Factor Ontology	13	1	1	6695	2.080	0.500	0.0	16411
O_{20}	Cell Ontology	16	18	2	7664	2.102	0.661	0	9821
O_{21}	Human Phenotype Ontology	16	14	2	6018	2.423	0.617	0.0007	9129
O_{24}	Nano particle Ontology	19	45	5	5176	2.329	0.500	0.0556	4267
O_{26}	Non-codingRNA	16	29	3	2418	1.450	0.584	0.0	5323
O_{28}	Statistic Ontology	9	5	1	673	1.776	0.594	0.0004	2310
O_{29}	Neural Electromagnetic Ontology	21	75	6	2644	2.244	0.5	0	4502
O_{30}	New Born Ontology	4	3	1	1048	1.738	0.685	0	3719
O_{32}	Animal trait ontology	11	8	2	83	1.808	0.500	0.0	2097
O_{33}	Ontology of Pneumology	13	10	4	7	1.398	0.500	0	1175
O_{34}	Metagenome and Microbiology Ontology	6	1	1	5	1.283	0.500	0.0	796
O_{35}	Human Physiology simulation ontology	13	11	1	1368	2.051	0.553	0.0002	4235
O_{36}	Sleep Domain Ontology	19	10	3	1363	2.065	0.597	0.0037	2826
O_{40}	Environment ontology for livestock	10	9	2	15	1.797	0.500	0.0	651
O_{42}	Human dermatological disease Ontology	12	12	2	359	2.155	0.551	0.0	1056
O_{43}	Cognitive Atlas Ontology	6	3	1	2507	1.560	0.609	0.0	4105
O_{44}	Cell type ontology	5	4	1	2	0.528	0.5	0	1710
O_{45}	Ontology of physics for biology	10	9	2	114	1.882	0.528	0.0	951
O_{46}	Ontology of MicroRNA Target	35	27	3	569	2.421	0.559	0.0003	4449
O_{48}	Adult mouse brain	5	3	1	1	1.717	0.500	0.0387	915
O_{49}	Ontology of biological and clinical statistic	12	5	3	617	2.177	0.549	0.0080	1305
O_{50}	Radio oncology ontology	11	5	3	352	1.777	0.550	0.0	1945
O_{53}	Emotion Ontology	9	6	2	234	2.258	0.580	0.0	509
O_{54}	Neuroscience Ontology	4	5	1	92	1.840	0.561	0.0	304
O_{55}	Neuroscience Information Ontology	14	7	4	210	1.855	0.518	0.0027	414
O_{56}	Ontology of genetic interval	7	13	1	575	1.903	0.579	0.0218	544
O_{57}	Population and Community Ontology	18	1	1	907	2.813	0.502	0.0025	211
O_{60}	Experimental Ontology	19	19	5	635	2.308	0.597	0.0025	1619
O_{61}	Immuno-genetics Ontology	13	8	2	88	2.219	0.565	0	3379
O_{63}	Brain Region Ontology	17	18	4	2741	2.362	0.597	0.0	1985
O_{66}	Fission Yeast Phenotype Ontology	7	4	1	715	2.047	0.609	0	582
O_{69}	Fanconi Anemia Ontology	4	6	1	124	1.205	0.533	0	1103
O_{70}	Medical image simulation	9	7	2	455	1.714	0.5	0	4530
O_{71}	Anatomical entity Ontology	21	133	4	307	2.004	0.539	0.0013	4243
O_{72}	Single-Nucleotide Polymorphism Ontology	11	6	2	121	2.031	0.56	0	261
O_{73}	HIV Ontology	14	86	3	110	2.057	0.572	0.0044	1401
O_{75}	Flora phenotype Ontology	12	9	2	332	1.823	0.572	0.0004	1239
O_{76}	Eagle resource research	13	9	1	1340	2.219	0.565	0	2333
O_{78}	Ontology of Drug Neuropathy adverse events	16	13	2	781	1.715	0.526	0.0010	4151
O_{79}	Neural-Immune Gene Ontology	22	48	6	1533	2.304	0.558	0.0004	1643
O_{80}	Kinetic simulation algorithm ontology	6	7	1	495	2.014	0.627	0.0	256
O_{82}	Sequence phenotype ontology	14	45	2	27059	1.700	0.604	0.00001	78116
O_{84}	Drug Interaction Knowledge Base Ontology	4	37	1	249	1.923	0.648	0.0249	298
O_{85}	Cell line Ontology	14	50	5	39915	2.563	0.0603	0.000001	11482
O_{86}	Breast Cancer Ontology	7	2	1	191	1.974	0.656	0.207	338
O_{68}	Autism spectrum ontology	6	5	1	1	1.489	0.500	0.0	284
O_{89}	Infectious Disease Ontology	9	5	1	284	2.083	0.596	0	373
O_{90}	Translational medicine ontology	11	10	2	47	2.128	0.534	0.0054	240
O_{91}	Ecosystem ontology	6	2	1	52	1.758	0.537	0.0	150
O_{94}	Symptom Ontology	7	3	1	98	1.691	0.500	0	937
O_{95}	Cancer Management and research ontology	18	7	3	141	1.866	0.558	0.0035	683
O_{96}	Biomedical Resource Ontology	7	5	1	20	1.540	0.500	0.0	528
O_{97}	Growth medium ontology	5	3	1	2	1.194	0.500	0.0	990
O_{98}	Epidemiology Ontology	9	8	3	57	2.131	0.508	0.0	195
O_{99}	Ontology of clinical research	6	3	1	257	1.877	0.639	0.0193	628
O_{100}	Mental State Assessment	11	76	3	2427	1.354	0.613	0.0035	1056

APPENDIX B: Normalised Decision matrix for WSM, WPM and WLCRT

Index	DIT	ANP	APL	TIP	EOG	RR	CR	SOV
O_1	0.2806	0.1364	0.42	0.1063	0.6301	0.7937	0.1085	0.1058
O_2	0.2806	0.1545	0.26	0.1111	0.5506	0.7514	0.18	0.1157
O_3	0.229	0.1242	0.18	0.1698	0.5369	0.7835	0.1015	0.1457
O_4	0.5387	0.5485	0.82	0.1016	0.7008	0.6631	0.18	0.2435
O_5	0.18	0.1121	0.18	0.1393	0.5236	0.82	0.18	0.1208
O_7	0.4613	0.2818	0.26	0.1122	0.2793	0.7079	0.18	0.1414
O_8	0.6161	0.1788	0.26	0.2691	0.6899	0.8308	0.18	0.588
O_9	0.2806	0.1485	0.26	0.1083	0.628	0.6759	0.18	0.1246
O_{11}	0.4871	0.3606	0.74	0.1397	0.7305	0.6631	0.3149	0.3239
O_{12}	0.1516	0.1182	0.26	0.82	0.6626	0.69	0.1162	0.82
O_{13}	0.229	0.1242	0.18	0.1056	0.4785	0.6989	0.18	0.1204
O_{14}	0.5645	0.3909	0.82	0.1209	0.7218	0.7374	0.1015	0.1165
O_{16}	0.4097	0.5242	0.42	0.5162	0.7242	0.8129	0.18	0.401
O_{17}	0.229	0.1182	0.18	0.1895	0.6266	0.8155	0.1003	0.1752
O_{18}	0.3323	0.18	0.18	0.1912	0.6434	0.6631	0.18	0.2159
O_{20}	0.4097	0.203	0.26	0.2044	0.6511	0.8693	0.18	0.1689
O_{21}	0.4097	0.1788	0.26	0.1819	0.7635	0.8129	0.1027	0.164
O_{24}	0.4871	0.3667	0.74	0.1705	0.7305	0.6631	0.3149	0.1293
O_{26}	0.4097	0.2697	0.42	0.1329	0.4228	0.7707	0.18	0.1369
O_{28}	0.229	0.1242	0.18	0.1092	0.5369	0.7835	0.1015	0.1154
O_{29}	0.5387	0.5485	0.82	0.136	0.7008	0.6631	0.18	0.131
O_{30}	0.18	0.1121	0.18	0.1143	0.5236	0.82	0.18	0.1254
O_{32}	0.2806	0.1424	0.26	0.1011	0.5481	0.6631	0.18	0.1139
O_{33}	0.3323	0.1545	0.58	0.1001	0.4046	0.6631	0.18	0.1073
O_{34}	0.1516	0.18	0.18	0.1001	0.3643	0.6631	0.18	0.1046
O_{35}	0.3323	0.1606	0.18	0.1186	0.6332	0.731	0.1008	0.1291
O_{36}	0.4871	0.1545	0.42	0.1185	0.6381	0.7873	0.1143	0.1191
O_{40}	0.2548	0.1485	0.26	0.1002	0.5443	0.6631	0.18	0.1036
O_{42}	0.3065	0.1667	0.26	0.1049	0.6696	0.7284	0.18	0.1065
O_{43}	0.1516	0.1121	0.18	0.1341	0.4613	0.8027	0.18	0.1282
O_{44}	0.1258	0.1182	0.18	0.1	0.18	0.6631	0.18	0.1111
O_{45}	0.2548	0.1485	0.26	0.1015	0.574	0.6989	0.18	0.1057
O_{46}	0.82	0.2576	0.42	0.1077	0.7628	0.7386	0.1012	0.1306
O_{48}	0.1258	0.1121	0.18	0.18	0.5163	0.6631	0.2496	0.1055
O_{49}	0.3065	0.1242	0.42	0.1084	0.6773	0.7258	0.1309	0.1082
O_{50}	0.2806	0.1242	0.42	0.1048	0.5373	0.7271	0.18	0.1128
O_{53}	0.229	0.1303	0.26	0.1032	0.7057	0.7655	0.18	0.1026
O_{54}	0.18	0.1242	0.18	0.1012	0.5593	0.7412	0.18	0.1011
O_{55}	0.3581	0.1364	0.58	0.1028	0.5646	0.6861	0.1104	0.1019
O_{56}	0.1774	0.1727	0.18	0.1078	0.5814	0.7643	0.1843	0.1028
O_{57}	0.4613	0.18	0.18	0.1123	0.82	0.6656	0.1097	0.1004
O_{60}	0.4871	0.2091	0.74	0.1086	0.7232	0.7873	0.1097	0.1105
O_{61}	0.3323	0.1424	0.26	0.1012	0.692	0.7463	0.18	0.123
O_{63}	0.4355	0.203	0.58	0.1373	0.7421	0.7873	0.18	0.1131
O_{66}	0.1774	0.1182	0.18	0.1097	0.6318	0.8027	0.18	0.1031
O_{69}	0.18	0.1303	0.18	0.1017	0.337	0.7053	0.18	0.1068
O_{70}	0.229	0.1364	0.26	0.1062	0.5152	0.6631	0.18	0.1312
O_{71}	0.5387	0.82	0.58	0.1042	0.6168	0.713	0.105	0.1292
O_{72}	0.2806	0.1303	0.26	0.1016	0.6262	0.7399	0.18	0.1008
O_{73}	0.3581	0.6152	0.42	0.1015	0.6353	0.7553	0.117	0.1089
O_{75}	0.3065	0.1485	0.26	0.1045	0.5534	0.7553	0.1015	0.1078
O_{76}	0.3323	0.1485	0.18	0.1182	0.692	0.7463	0.18	0.1156
O_{78}	0.4097	0.1727	0.26	0.1106	0.5156	0.6964	0.1039	0.1285

O_{79}	0.5645	0.3848	0.82	0.1209	0.7218	0.7374	0.1015	0.1106
O_{80}	0.1516	0.1364	0.18	0.1067	0.6203	0.8257	0.18	0.1008
O_{82}	0.3581	0.3667	0.26	0.4685	0.5103	0.7963	0.1	0.6558
O_{84}	0.18	0.3182	0.18	0.1034	0.5884	0.8526	0.1962	0.1011
O_{85}	0.3581	0.397	0.74	0.6436	0.8125	0.18	0.1	0.1808
O_{86}	0.1774	0.1061	0.18	0.1026	0.6063	0.8629	0.82	0.1013
O_{68}	0.1516	0.1242	0.18	0.18	0.4365	0.6631	0.18	0.101
O_{89}	0.229	0.1242	0.18	0.1039	0.6444	0.786	0.18	0.1016
O_{90}	0.2806	0.1545	0.26	0.1006	0.6602	0.7066	0.1209	0.1006
O_{91}	0.1516	0.1061	0.18	0.1007	0.5306	0.7105	0.18	0.18
O_{94}	0.1774	0.1121	0.18	0.1013	0.5072	0.6631	0.18	0.1056
O_{95}	0.4613	0.1364	0.42	0.1019	0.5684	0.7374	0.1135	0.1038
O_{96}	0.1774	0.1242	0.18	0.1003	0.4543	0.6631	0.18	0.1027
O_{97}	0.1258	0.1121	0.18	0.1	0.3332	0.6631	0.18	0.106
O_{98}	0.229	0.1424	0.42	0.1008	0.6612	0.6733	0.18	0.1003
O_{99}	0.1516	0.1121	0.18	0.1035	0.5723	0.8411	0.1746	0.1034
O_{100}	0.2806	0.5545	0.42	0.133	0.3892	0.8078	0.1135	0.1065

APPENDIX C: Normalised Decision matrix for TOPSIS and ELECTRE

Index	DIT	ANP	APL	TIP	EOG	RR	CR	SOV
O_1	0.0223	0.0124	0.0309	0.0051	0.0179	0.0042	0.0019	0.0039
O_2	0.0223	0.0141	0.0191	0.0053	0.0156	0.004	0.0031	0.0043
O_3	0.0182	0.0113	0.0132	0.0081	0.0152	0.0042	0.0017	0.0054
O_4	0.0428	0.05	0.0603	0.0049	0.0199	0.0035	0.0031	0.009
O_5	0.0143	0.0102	0.0132	0.0067	0.0149	0.0044	0.0031	0.0045
O_7	0.0367	0.0257	0.0191	0.0054	0.0079	0.0038	0.0031	0.0052
O_8	0.049	0.0163	0.0191	0.0129	0.0196	0.0044	0.0031	0.0218
O_9	0.0223	0.0135	0.0191	0.0052	0.0178	0.0036	0.0031	0.0046
O_{11}	0.0387	0.0329	0.0544	0.0067	0.0207	0.0035	0.0054	0.012
O_{12}	0.012	0.0108	0.0191	0.0393	0.0188	0.0037	0.002	0.0304
O_{13}	0.0182	0.0113	0.0132	0.0051	0.0136	0.0037	0.0031	0.0045
O_{14}	0.0449	0.0356	0.0603	0.0058	0.0205	0.0039	0.0017	0.0043
O_{16}	0.0326	0.0478	0.0309	0.0248	0.0206	0.0043	0.0031	0.0149
O_{17}	0.0182	0.0108	0.0132	0.0091	0.0178	0.0043	0.0017	0.0065
O_{18}	0.0264	0.0164	0.0132	0.0092	0.0183	0.0035	0.0031	0.008
O_{20}	0.0326	0.0185	0.0191	0.0098	0.0185	0.0046	0.0031	0.0063
O_{21}	0.0326	0.0163	0.0191	0.0087	0.0217	0.0043	0.0018	0.0061
O_{24}	0.0387	0.0334	0.0544	0.0082	0.0207	0.0035	0.0054	0.0048
O_{26}	0.0326	0.0246	0.0309	0.0064	0.012	0.0041	0.0031	0.0051
O_{28}	0.0182	0.0113	0.0132	0.0052	0.0152	0.0042	0.0017	0.0043
O_{29}	0.0428	0.05	0.0603	0.0065	0.0199	0.0035	0.0031	0.0049
O_{30}	0.0143	0.0102	0.0132	0.0055	0.0149	0.0044	0.0031	0.0046
O_{32}	0.0223	0.013	0.0191	0.0048	0.0156	0.0035	0.0031	0.0042
O_{33}	0.0264	0.0141	0.0427	0.0048	0.0115	0.0035	0.0031	0.004
O_{34}	0.012	0.0164	0.0132	0.0048	0.0103	0.0035	0.0031	0.0039
O_{35}	0.0264	0.0146	0.0132	0.0057	0.018	0.0039	0.0017	0.0048
O_{36}	0.0387	0.0141	0.0309	0.0057	0.0181	0.0042	0.002	0.0044
O_{40}	0.0203	0.0135	0.0191	0.0048	0.0155	0.0035	0.0031	0.0038
O_{42}	0.0244	0.0152	0.0191	0.005	0.019	0.0039	0.0031	0.0039
O_{43}	0.012	0.0102	0.0132	0.0064	0.0131	0.0043	0.0031	0.0047
O_{44}	0.01	0.0108	0.0132	0.0048	0.0051	0.0035	0.0031	0.0041

O_{45}	0.0203	0.0135	0.0191	0.0049	0.0163	0.0037	0.0031	0.0039
O_{46}	0.0652	0.0235	0.0309	0.0052	0.0217	0.0039	0.0017	0.0048
O_{48}	0.01	0.0102	0.0132	0.0086	0.0147	0.0035	0.0043	0.0039
O_{49}	0.0244	0.0113	0.0309	0.0052	0.0192	0.0039	0.0022	0.004
O_{50}	0.0223	0.0113	0.0309	0.005	0.0153	0.0039	0.0031	0.0042
O_{53}	0.0182	0.0119	0.0191	0.0049	0.02	0.0041	0.0031	0.0038
O_{54}	0.0143	0.0113	0.0132	0.0049	0.0159	0.004	0.0031	0.0037
O_{55}	0.0285	0.0124	0.0427	0.0049	0.016	0.0037	0.0019	0.0038
O_{56}	0.0141	0.0157	0.0132	0.0052	0.0165	0.0041	0.0032	0.0038
O_{57}	0.0367	0.0164	0.0132	0.0054	0.0233	0.0035	0.0019	0.0037
O_{60}	0.0387	0.0191	0.0544	0.0052	0.0205	0.0042	0.0019	0.0041
O_{61}	0.0264	0.013	0.0191	0.0049	0.0196	0.004	0.0031	0.0046
O_{63}	0.0346	0.0185	0.0427	0.0066	0.0211	0.0042	0.0031	0.0042
O_{66}	0.0141	0.0108	0.0132	0.0053	0.0179	0.0043	0.0031	0.0038
O_{69}	0.0143	0.0119	0.0132	0.0049	0.0096	0.0038	0.0031	0.004
O_{70}	0.0182	0.0124	0.0191	0.0051	0.0146	0.0035	0.0031	0.0049
O_{71}	0.0428	0.0747	0.0427	0.005	0.0175	0.0038	0.0018	0.0048
O_{72}	0.0223	0.0119	0.0191	0.0049	0.0178	0.0039	0.0031	0.0037
O_{73}	0.0285	0.056	0.0309	0.0049	0.018	0.004	0.002	0.004
O_{75}	0.0244	0.0135	0.0191	0.005	0.0157	0.004	0.0017	0.004
O_{76}	0.0264	0.0135	0.0132	0.0057	0.0196	0.004	0.0031	0.0043
O_{78}	0.0326	0.0157	0.0191	0.0053	0.0146	0.0037	0.0018	0.0048
O_{79}	0.0449	0.0351	0.0603	0.0058	0.0205	0.0039	0.0017	0.0041
O_{80}	0.012	0.0124	0.0132	0.0051	0.0176	0.0044	0.0031	0.0037
O_{82}	0.0285	0.0334	0.0191	0.0225	0.0145	0.0042	0.0017	0.0243
O_{84}	0.0143	0.029	0.0132	0.005	0.0167	0.0045	0.0034	0.0037
O_{85}	0.0285	0.0362	0.0544	0.0309	0.0231	0.001	0.0017	0.0067
O_{86}	0.0141	0.0097	0.0132	0.0049	0.0172	0.0046	0.014	0.0038
O_{68}	0.012	0.0113	0.0132	0.0086	0.0124	0.0035	0.0031	0.0037
O_{89}	0.0182	0.0113	0.0132	0.005	0.0183	0.0042	0.0031	0.0038
O_{90}	0.0223	0.0141	0.0191	0.0048	0.0187	0.0038	0.0021	0.0037
O_{91}	0.012	0.0097	0.0132	0.0048	0.0151	0.0038	0.0031	0.0067
O_{94}	0.0141	0.0102	0.0132	0.0049	0.0144	0.0035	0.0031	0.0039
O_{95}	0.0367	0.0124	0.0309	0.0049	0.0161	0.0039	0.0019	0.0038

O_{96}	0.0141	0.0113	0.0132	0.0048	0.0129	0.0035	0.0031	0.0038
O_{97}	0.01	0.0102	0.0132	0.0048	0.0095	0.0035	0.0031	0.0039
O_{98}	0.0182	0.013	0.0309	0.0048	0.0188	0.0036	0.0031	0.0037
O_{99}	0.012	0.0102	0.0132	0.005	0.0162	0.0045	0.003	0.0038
O_{100}	0.0223	0.0505	0.0309	0.0064	0.011	0.0043	0.0019	0.0039

APPENDIX D: Dominance Matrixes

1 1 0 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 1 1 1 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1
0 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 1 0 0 0 1 1 1 0 0 0 0 1 0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 0 1 1 1 0 1 0 1 0 1 0 0
0 0 0 1 1 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 1 1 0 0 1 0 0 1 1 0 1 0 0 0 0 1 1 1 0 0 0 0 0 1 0 1 0 1 0 1 1 1 0 1 1 0 1 1 0 1 0
1 1 0 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 1 0 1 1 1 1 1 0 1 0 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 1 1 1 0 1 0 1 0 1 1 1 1 0 1 0 1 1 0 1 0
1 0 0 0 0 0 1 0 1 1 0 0 0 0 1 1 0 0 1 0 0 1 1 0 0 0 1 1 0 0 1 0 0 1 1 1 1 1 1 0 0 1 0 1 1 1 0 1 0 1 0 1 0 1 0 0 0 1 0 1 1 0 1 1 1 0 1 1 0
1 1 0 1 1 0 1 0 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1
1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 1 0 0 0 1 1 1 0 0 0 0 1 0 0 1 1 0 1 0 1 0 0 0 1 0 0 0 0 1 1 1 0 1 0 1 0 1 0 0
1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 1 0 1 1 0 1 1 0 1 0
1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 0 1 1 0 1 0 0 0 0 1 1 1 0 1 0 1 0 1 0 1 0 1 0 1 1 1 0 1 1 0 1 1 0 1 0
1 1 0 1 0 0 1 0 1 1 0 0 1 0 0 1 0 0 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 0 1 0 1 1 1 0 1 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 0
1 1 0 1 1 0 1 0 1 1 0 0 1 1 0 1 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 0 1 1 1 0 1 1 1 1 1 0 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 0
1 1 0 1 1 1 1 0 1 1 0 0 1 0 1 0 0 0 1 0 1 1 1 0 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 0 1 1 1 0 1 0 1 1 1 0 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 0
1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 0 0 1 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1
0 1 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1 0 0 1 0 0 1 1 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 1 0 1 1 0 1 1 0 1 0
1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
1 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1 1 1 1 0 1 0 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 1 1 1 0 1 0 1 0 1 1 1 1 0 1 0 1 1 0 1 0
0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0
1 1 0 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 0 1 1 0 1 1 0 1 1 1 1 1 0 0 0 0 1 0 1 1 1 0 1 0 1 1 0 0 1 0 0 0 0 1 1 1 0 1 0 1 1 1 1 1
0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 1 1 0 0 1 0 0 1 1 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 1 0 1 0 1 1 1 1 1 1 0 1 1 0 1 0
1 1 0 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 0 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0 1 1 1 0 1 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 1 0 1 0
1 1 0 1 1 0 1 0 1 1 0 0 1 0 0 0 0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0
1 1 0 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 0 1 1 1 0 1 0 1 0 1 0 0
1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 0 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 1 0 1 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 1 1 0 0 0
0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 0 0
1 1 0 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 1 1 0 0 0 0 0 1 1 0 1 0 0 0 0 1 1 1 0 1 0 0 0 1 0 1 0 1 0 1 1 1 1 0 1 0 1 1 0 1 0
1 1 0 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 0 1 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1 1 0 1 0 0 0 0 1 1 1 0 1 0 1 0 0 0 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1
0 1 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 1 0 1 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1
0 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0
0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0
1 1 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 1 1 1 0 0 1 0 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 1 0 1 0 1 0 0 0 1 1 0 0 1 1 1 1 1 1 0 1 1 1 0 1
1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1 0 1 1 0 0 0 0 1 1 0 0 0 0 1 0 1 1 1 0 1 0 1 0 1 0 1 0 0 0 1 1 1 1 0 1 0 1 0 0 0 0
1 1 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 0 1 1 1 0 1 0 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0
1 1 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 0 0 1 1 0 0 0 1 1 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 0 1 1 1 0 0 0 1 0 1 0 0
1 1 0 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0
0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 0 0 0 0
1 1 0 1 1 0 1 0 1 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0
1 1 0 1 1 0 1 0 1 1 0 0 1 1 0 0 0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1 0 1 1 1 0 1 0 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1 0 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0
1 1 0 1 0 0 1 0 1 1 0 0 1 1 0 0 0 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 0 1 0 0 1 0 1 1 1 0 1 0 1 0 0 0 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 1 0
1 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 1 1 0 1 1 0 0 1 1 1 1 1 1 0 0 1 0 1 1 1 0 1 0 1 0 0 0 1 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0
1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
0 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 1 1 0 0 1 0 0 1 1 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 1 0
1 1 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 0 1 1 1 0 1 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 0
1 1 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 1 0 1 0 1 0 0 0 1 1 0 1 1 1 0 1 1 0 1 0
1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 1 1 0 1 0 1 0 1 0 0 0 0 0 0 1 0 1 1 1 0 1 1 0 1 0
0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 0
0 1 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 1 1 0 0 1 0 0 1 1 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 1 0 1 0 1 1 0 0 1 1 0 1 1 0 1 0
1 1 0 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0
0 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 0 0 1 0 1 0 0 0 0 1 1 1 0 0 0 0 0 1 0 1 0 0 0 1 1 0 1 0 1 0 1 1 0 1 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 0
1 1 0 1 1 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 1 1 0 1 0 0 1 0 1 1 1 0 1 1 1 1 1 1 0 0 0 0 0 1 1 1 0 1 0 1 0 1 0 1 1 0 0 0 1 1 1 1 1 0 1 1 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 1 1 0 1 0 0 1 1 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 1 0 1 0 1 1 1 1 0 1 0 1 0 0 1 0
0 1 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 0 1 0 1 1 1 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 1 0 1 0 1 1 1 0 1 1 1 1 1 0 1 0
1 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 0 1 0 0 1 1 0 1 0 0 0 0 1 1 1 0 1 0 1 0 1 0 1 0 1 0 1 1 0 1 1 1 0 1 1 0 0 0
1 1 0 1 1 0 1 0 1 1 0 0 1 0 0 0 0 0 1 0 1 1 0 1 1 0 1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 0 0 1 1 1 0 1 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0

Concordance Dominance Matrix

$$
\begin{bmatrix}
110110110011110100100100000010100010110100010100110111100000100001 \\
111111111111111100101101000101110101111000111111101111011001001 01 \\
1111101111111110111111101100111110111111100111111111111011111101111 \\
00000001000000000000000000000000000000000000100000000010000000000000 \\
1111111111111111011011111011111101111111011111111111111011101001 11 \\
01111011011111010010110100010111010111100011111110101000100100101 \\
00010011000011010010010000010000010010100010100010101000000100001 \\
110111111111111100101101000101100101111000101111101111011001001 01 \\
00000010000000001000000000000000000000000000010100010001000000000000 \\
110100110011110100100100000100000101101000101001101110000000100001 \\
1111101111111111111111111110111111111111100111111111111111111101111 \\
00010001000000001000000000000000000000000000010100000001000000000000 \\
00010010000001001000000000000100000000010000010000010001000000000000 \\
1111101101111101111011111011111101111110001111111011110011 1100111 \\
110110110011110100100100000101100101101000101001101111000000100101 \\
010110110001110100100100000101100101101000101000101010000000100101 \\
010110110000110100100100000101100101101000101000101010000000100101 \\
00010010000000001000000000000000000000000000010100010001000000000000 \\
010110110001100100100100000101000101101000101010101010000000000001 \\
1111111111111110111111101100111110111111100111111111111111111101111 \\
00000000000000000000000000000000000000000000100000000010000000000000 \\
1111111111111111011011111011111101111111011111111111111111110100111 \\
111111111111111100101101000101110101111000111111101111011001001 01 \\
010110110000110100000010000010000010010100010100010001000000100001 \\
1111111111111111111101111101111111111111101111111111111101111100111 \\
110110110111110100100100000101100101101000101011101111000000100101 \\
110110110000110100100000000100000100101000101000101010000000000001 \\
1111111111111111011011011011111101011110001111111111111111110100111 \\
110110110111110100101100000101100101111000101011101111000000100101 \\
1111101111111110111101111001111111111111100111111111111011101001 11 \\
1111111111111111111111111110111111111111101111111111111111111111111 \\
1111111111111111011010110011111010111100011111111111111111110100111 \\
010110110000100100000000000000000000101000101000101010000000000001 \\
1111111111111111011011111011011101111111011111111111111011101001 01 \\
110110110000110100100100000100000101101000101000101111000000100001 \\
110110110011110100100100000101000101101000101001101111000000100101 \\
1111101111111110111101101000101101111111100111111111111011101001 11 \\
1111111111111111111111111110111111011111101111111111111111111100111 \\
010110110000110100000010000010000000001010001010001010100000000000 01 \\
1111111111111111011011111011111101011110011111111111111111110100111 \\
010110110011110100100100000101100100101000101000101010000000100001 \\
00010011000010010000000000000000000000101000100010000000000000001 \\
110110110111110100101101000101100101101000101011101111000000100101 \\
00011011000010010000000000001000000000100000101000100010000000000001 \\
1111111111111111111101111101111110111110011111111111111011111100111 \\
1111111111111111111111111111111111111111101111111111111111111111111 \\
1111111111111111111111111110111111111111100111111111111111111110111 \\
00000001000000001000000000000000000000000000000000000010000000000000 \\
1111111111111111001011011011011001011110001011111011110111010011 1 \\
00010011000010010000000000001000000000100000100000101010000000000000 \\
1111101101111101001011010001011101011110001110111011110010010010 1 \\
110110110111110100100100000101100101101000101001101111000000100101 \\
110110110111110100100100000101100101111000101010101110000000100101 \\
00010011000000001000000000000000000000000000010100000001000000000000 \\
1111101111111110111101111101111110111111000111111101111011111100111 \\
010100110000110100100100000101100100101000101000100010000000100101 \\
110110110111110100101101000101100101111000101111101011000000100101 \\
00000010000000001000000000001000000000000000010000010000000000000000 \\
1111101101111101011011110011011001011110001111111011100010010010 1 \\
1111101111111111111111111110111111111111101111111111111011111101111 \\
1111101111111110111101011001111100111110001111111111111001111100111 \\
111110111111111100101101000101100101111000101011101111010001001 01 \\
1111101111111110101101111101111110101111100111111111111111100100101 \\
1111111111111111111101111101111110111111011111111111111111110100111 \\
110110110000110100100100000100000100101000101000101110000000000001 \\
1111111111111111111111111111111111111111101111111111111111111101111 \\
1111111111111111111111111110111111111111101111111111111111111110111 \\
110110110111110100101100000101100101101000101011101111000000100001 \\
1111101111111110101101111001111110101111000111111101111101101001 01 \\
110110110000100100000000000010000000001000001010001010100000000000 00
\end{bmatrix}
$$

Discordance Dominance Matrix

0 0 0 0 1 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1 1 0 0 1 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 1 1 1 0 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 1 1 0 1 0 1 0 1 0 1 1 0 0 0 0 0 0 1 0
1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1
1 0 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 1 0 1 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 1 0 1 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 1 0
1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0
1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0
1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 1 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 1 0 1 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1 1 0 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 1 1 0 1 0 1 0 1 1 1 1 0 0 0 0 0 0 1 0
0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 1 0 0 0 1 0 0 1 1 0 1 0 0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 1 1 1 1 0 0 0 0 1 0
1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 1 0 0 1 0 1 0 0 1 1 0 1 0 0 1 0 1 0 0 0 1 0 1 1 1 0 1 0 1 0 1 0 1 1 1 0 0 0 0 1 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 1 0 0 0
0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1
0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 1 0 1 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0
1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1 0 0 0 0 1 0
1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 0
0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 1 1 0 0 0 0 1 0
1 0 1 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 1 0 1 0 0 1 1 0 1 0 0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1 0 1 1 1 1 0 1 0 1 0 0 1 0

Aggregate Dominance Matrix