

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Nicolas Ferranti

An experimental analysis on ontology meta-matching

Juiz de Fora

2020

Nicolas Ferranti

An experimental analysis on ontology meta-matching

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. D.Sc. Jairo Francisco de Souza

Coorientador: Prof. D.Sc Stênio Sã Rosário Furtado Soares

Juiz de Fora

2020

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF com os dados fornecidos pelo(a) autor(a)

Ferranti, Nicolas.

An experimental analysis on ontology meta-matching / Nicolas Ferranti.
– 2020.

118 f. : il.

Orientador: Prof. D.Sc. Jairo Francisco de Souza

Coorientador: Prof. D.Sc Stênio São Rosário Furtado Soares

Dissertação (mestrado acadêmico) – Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação, 2020.

1. Ontology. 2. Ontology Matching. 3. Ontology Meta-Matching. 4. Semantic Web. 5. Optimization Metaheuristics. I. de Souza, Jairo Francisco, orient. II. Título.



Nicolas Ferranti

“An experimental analysis on ontology meta-matching”

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre em Ciência da Computação.

Aprovada em 28 de agosto de 2020.

BANCA EXAMINADORA

Prof. Dr. Jairo Francisco de Souza – Orientador
Universidade Federal de Juiz de Fora

Prof. Dr. Stênio Sã Rosário Furtado Soares - Coorientador
Universidade Federal de Juiz de Fora

Prof. Dr. Carlos Cristiano Hasenclever Borges
Universidade Federal de Juiz de Fora

Profª. Dra. Kate Cerqueira Revoredo
Vienna University of Economics and Business

AGRADECIMENTOS

First of all, I would like to thank my family, especially my parents, Silvana and Antônio, for all the love and support that were essential for this moment.

I would like to thank my brother Tomás, for always providing me with the best discussions and debates on computational problems.

To the companions who collaborated with this work, in particular, José Ronaldo Mouro, João Victor Souza, and Vinicius Alberto Alves da Silva.

To my friends, my thanks for all the support and fun moments. Without you, this whole process would be much more difficult.

I am also grateful to the professors at DCC and PPGCC who contributed to my academic education through their valuable teachings and advice. I am especially grateful to my advisor and friend Jairo Francisco de Souza for his guidance and partnership over the years, from undergraduate to master's degree.

To my co-advisor Stênio Sã, for accepting to participate in this journey and contributing to the development of this work.

To Professor Fabrício, for his contribution in the development of the data provenance model for ontology meta-matching.

Finally, I would like to thank all ICE employees who have always helped me.

“Seja você quem for, seja qual for a posição social que você tenha na vida, a mais alta ou a mais baixa, tenha sempre como meta muita força, muita determinação e sempre faça tudo com muito amor e com muita fé em Deus, que um dia você chega lá. De alguma maneira você chega lá.”

Ayrton Senna

RESUMO

A interoperabilidade semântica é um dos principais objetivos a serem alcançados por sistemas distribuídos e abertos baseados em conhecimento. Atualmente, a tecnologia mais adequada para realizar essa tarefa é representada por ontologias. No entanto, a grande capacidade de representação do modelo ontológico é dificultada pelo problema da heterogeneidade semântica que afeta o consumo de mais de uma ontologia na mesma aplicação. A solução mais usada para superar esse problema é encontrar um alinhamento entre as ontologias com base no processo de alinhamento de ontologias. Como o alinhamento de ontologias se tornou uma questão-chave para resolver problemas de heterogeneidade semântica, vários pesquisadores propuseram diversas técnicas que podem ser usadas em cenários distintos. Meta-alinhamento de ontologias é um subproblema relacionado ao processo de seleção dos algoritmos, pesos e alinhadores mais apropriados em diferentes cenários de alinhamento. Meta-alinhadores de ontologias alcançaram bons resultados em pares de ontologias com diferentes tipos de heterogeneidades. Embora surjam a cada ano novas abordagens para meta-alinhamento de ontologias, ainda há muito a aprender sobre os métodos existentes. Este trabalho apresenta uma análise experimental do problema de meta-alinhamento de ontologias. Primeiramente, foi realizado um mapeamento sistemático da literatura para entender a sequência de etapas necessárias para a construção de uma solução, além de como o problema pode ser modelado e quais são as principais técnicas utilizadas pelos pesquisadores. Os resultados desse mapeamento foram utilizados na construção de um framework para simplificar a experimentação científica em meta-alinhamento de ontologias. Nesse contexto, foi possível estudar o comportamento e as características de várias técnicas comumente usadas em cada etapa do workflow. Os resultados mostraram que, quanto mais características das entidades puderem ser capturadas pelo conjunto de medidas de similaridade, maior a acurácia do modelo. Também foi possível observar o bom desempenho e acurácia das meta-heurísticas baseadas em otimização local quando comparadas às meta-heurísticas de otimização global. Experimentos com diferentes funções objetivo mostraram que métodos semi-supervisionados podem reduzir o tempo de execução do experimento, mas, por outro lado, trazem maiores variações no alinhamento resultante. 2

Palavras-chave: Meta-alinhamento de Ontologias. Ontologias. Meta-heurísticas. Web Semântica. Aprendizado Supervisionado. Aprendizado Não Supervisionado.

ABSTRACT

Semantic interoperability is one of the main objectives to be achieved by distributed and open knowledge based systems. Nowadays, the most appropriate technology to accomplish this task is represented through ontologies. However, the great representation capacity of the ontological model is hindered by the problem of semantic heterogeneity that affects the consumption of more than one ontology in the same application. The most used solution to overcome this problem is to find an alignment between the ontologies based on the ontology matching process. As ontology matching has become a key issue to solve semantic heterogeneity problems, several researchers have proposed diverse techniques that can be used in distinct scenarios. Ontology meta-matching is a sub-problem related to the selection process of the most appropriate algorithms, matchers and weights in different matching scenarios. Ontology meta-matching approaches have achieved good results in pairs of ontologies with different types of heterogeneities. Although new approaches to ontology meta-matching emerge each year, there is still a lot to learn about existing methods. This work presents an experimental analysis on the ontology meta-matching problem. First, a systematic mapping of the literature was conducted in order to understand the sequence of steps necessary to build a solution, furthermore, how the problem can be modeled, and what are the main techniques used by the researchers. The results of this mapping were used in the construction of a framework to simplify scientific experimentation in ontology meta-matching. In this context, a study was carried out to analyze the behavior and characteristics of several techniques commonly used at each stage of the workflow. Results showed that, the more characteristics of the entities that can be captured by similarity measures set, the greater the accuracy of the model. It was also possible to observe the good performance and accuracy of local search based meta-heuristics when compared to global optimization meta-heuristics. Experiments with different objective functions have shown that semi-supervised methods can shorten the execution time of the experiment but, on the other hand, bring greater variations in the resulting alignment.

Key-words: Ontology Meta-Matching. Ontologies. Meta-heuristics. Semantic Web. Supervised Learning. Unsupervised Learning.

LIST OF FIGURES

Figure 1 – Schematic of steps taken in the research conduction divided by chapters . . .	24
Figure 2 – Classification of the similarity measures	29
Figure 3 – Systematic mapping conduction	35
Figure 4 – Number of mono-objective and multi-objective approaches among the articles found by the systematic mapping	35
Figure 5 – Total articles found in the systematic mapping divided by country	36
Figure 6 – Citations graph between the articles of the systematic mapping	38
Figure 7 – Citations graph explanation	39
Figure 8 – Percentage of each type of similarity measure per year	42
Figure 9 – Number of articles using weight-based and correspondence-based modeling	48
Figure 10 – Mono-objective/multi-objective and supervised/unsupervised approaches over the years.	50
Figure 11 – System architecture, representing the execution flow that starts in the Manager and ends in the analysis of the data using the Statistics module. The dashed line represents the provenance data flow while the continuous line represents the flow of the data that will be consumed for the alignment construction. Orange colored blocks represent the processing modules, responsible for carrying out the activities that will create the final alignment.	62
Figure 12 – Data Flow of the meta-matching process, showing the confidence calculated by each similarity measure for each pair of entities, the aggregated confidence calculated by the Optimizer, and the alignment chosen by the Correspondence Selector module	70
Figure 13 – Starting classes of the PROV-O ontology	73
Figure 14 – Provenance Model for Ontology Meta-Matching. The model starts by captu- ring the definition of the experiment, followed by the calculation of weights, calculation of the similarity matrix, definition of the alignment and, finally, the alignment evaluation.	73
Figure 15 – Provenance RDF-based Output File	74
Figure 16 – Configurations created for evaluation. Each yellow container represents a processing module, the white balloons represent the algorithms used in each module. The name of the configurations used in the experiments are in the grey rectangle.	75
Figure 17 – Mean time in seconds for all experiments in all test cases	78
Figure 18 – Mean F-measure (horizontal line) and standard deviation of the mean F- measure samples (vertical line) for each experiment model for each test case	79
Figure 19 – Percentage of each type of similarity measure per year	85
Figure 20 – Amount of Unsupervised vs Supervised/Semi-supervised approaches per year	86

Figure 21 – System Architecture	87
Figure 22 – Data Flow	91
Figure 23 – Configurations created for evaluation of H1	92
Figure 24 – Configurations created for evaluation of H2	93
Figure 25 – Configurations created for evaluation of H3	94
Figure 26 – Configurations created for evaluation of H4	94
Figure 27 – Best F-measure rate for each H1 model, for each test case	97
Figure 28 – Mean runtime in seconds of each model for each test case	97
Figure 29 – Mean runtime (seconds) of each H2 model for each test case	99
Figure 30 – Mean runtime (seconds) of each H3 model for each test case	101
Figure 31 – Mean F-measure and deviation of H3 linear system based experiment models	102
Figure 32 – Mean F-measure and deviation of MatchFm based experiment models . . .	103
Figure 33 – Mean runtime (seconds) of each H4 model for each test case	104

LIST OF TABLES

Table 1 – PICO Elements	32
Table 2 – PICO Keywords	32
Table 3 – Amount of retrieved papers from each repository	34
Table 4 – Authors’ names and number of publications	36
Table 5 – Publishing Vehicle Types	39
Table 6 – Most Popular Journals	39
Table 7 – Most Popular Conferences	40
Table 8 – Most commonly used similarity measures grouped by technique classification	40
Table 9 – Metaheuristics used in OMM literature	43
Table 10 – Representation models	47
Table 11 – OAEI Datasets	50
Table 12 – F-measure values in OAEI datasets.	52
Table 13 – Similarity Measures Available	63
Table 14 – Genetic Algorithm Configuration	75
Table 15 – GRASP Configuration	76
Table 16 – Precision, Recall, and F-measure rates for the four experiment models	76
Table 17 – Jaccard, Overlap, and Dice Coefficients for test number 301	77
Table 18 – Similarity Measures	88
Table 19 – Objective Functions Available	90
Table 20 – F-measure rates for the four experiment models	96
Table 21 – Jaccard Coefficient for H1 test number 209	96
Table 22 – Jaccard Coefficient for H1 test number 258	96
Table 23 – Comparison of SBL + TB with literature approaches in terms of F-measure	98
Table 24 – F-measure rates for the four experiment models of H2	99
Table 25 – F-measure rates for the six experiment models of H3	100
Table 26 – F-measure rates for the two experiment models of H4	104
Table 27 – List of papers used in Figure 6	109

LIST OF ACRONYMS

CS	Computer Science
GA	Genetic Algorithm
GRASP	Greedy Randomized Adaptive Search Procedure
MOEA/D	Multi-objective Evolutionary Algorithm based on Decomposition
NSGA-II	Non-dominated Sorting Genetic Algorithm-II
OAEI	Ontology Alignment Evaluation Initiative
OM	Ontology Matching
OMM	Ontology Meta-matching
OMOPSO	Optimal Multi-Objective Particle Swarm Optimization
PICO	Population Intervention Comparison and Outcomes
PPA	Prey-Predator Algorithm
PSO	Particle Swarm Optimization
SA	Simulated Annealing
SPEA2	Strength Pareto Evolutionary Algorithm 2

SUMMARY

1	INTRODUCTION	20
1.1	OBJECTIVES	21
1.2	ONTOLOGY META-MATCHING	21
1.3	CHAPTERS OVERVIEW	23
1.4	CONTRIBUTIONS	24
2	MANUSCRIPT 1: LITERATURE REVIEW	26
2.1	INTRODUCTION	26
2.2	ONTOLOGY MATCHING	27
2.3	RELATED WORKS	28
2.3.1	Ontology matching reviews	29
2.3.2	Ontology meta-matching reviews	30
2.4	RESEARCH PROTOCOL	31
2.5	MAPPING REPORT	34
2.5.1	RQ1: How many studies were published over the years?	34
2.5.2	RQ2: Who are the most active authors in the area?	36
2.5.3	RQ3: Which publication vehicles are the main targets for research production in the area?	37
2.5.4	RQ4: What are the most common similarity measures used?	39
2.5.5	RQ5: What are the main optimization meta-heuristics used?	41
2.5.6	RQ6: How have solutions been represented?	46
2.5.7	RQ7: What are the main objective functions?	48
2.5.8	RQ8: Which datasets has been used for OMM evaluation?	50
2.6	THREATS TO VALIDITY	53
2.7	CONCLUDING REMARKS	53
3	MANUSCRIPT 2: A FRAMEWORK FOR OMM EVALUATION	55
3.1	INTRODUCTION	55
3.2	PROBLEM STATEMENT	56
3.3	RELATED WORK	58
3.4	PROPOSED APPROACH	61
3.4.1	Manager	62
3.4.2	Modeling	62
3.4.3	Optimization	63
3.4.4	Correspondence Selection	69
3.4.5	Statistics Generator	69

3.4.6	Provenance Collector	71
3.5	EVALUATION	73
3.6	CONCLUSIONS	77
4	MANUSCRIPT 3: EXPERIMENTAL RESULTS	80
4.1	INTRODUCTION	80
4.2	RELATED WORKS	81
4.3	HYPOTHESIS DEFINITION	84
4.4	ONTOLOGY META-MATCHING EXPERIMENTATION FRAMEWORK	87
4.4.1	Modeling	88
4.4.2	Optimization	89
4.4.3	Correspondence Selection	90
4.5	EXPERIMENTAL PLANNING	92
4.6	EXPERIMENTAL RESULTS	94
4.6.1	H1 Evaluation	95
4.6.2	H2 Evaluation	97
4.6.3	H3 Evaluation	98
4.6.4	H4 Evaluation	100
4.7	CONCLUDING REMARKS	102
5	CONCLUSIONS	106
5.1	LIMITATIONS	107
5.2	CONTRIBUTIONS	108
	REFERENCES	110

,

1 INTRODUCTION

The progress of information and communication technologies is revolutionizing the way people interact with each other. New software appears every day and the need for speed in the exchange of information keeps pace with technological development. In this context, the Web appears as a large knowledge base concentrating data published by different agents, human or non-human.

For many years the biggest problem faced by applications that aimed to consume and exchange this data was the lack of metadata and structures that would facilitate the semantic interoperability. According to (ACAMPORA et al., 2013a), semantic interoperability is the capability of two or more information systems to meaningfully and accurately put the exchanged data into a common understanding so as to produce useful outcomes. Semantic interoperability is a fundamental feature of all open knowledge based systems that, once achieved, enables machine interpretation, computable logic and inference.

As a result of the procedures provided by the Semantic Web (BERNERS-LEE et al., 2001), ontologies have become the best recognized technology for assigning well-defined semantic meaning to the data (ACAMPORA et al., 2013a). However, ontologies are developed by people and organizations with different purposes, which means that the way of expressing the semantics of the data may diverge either in nomenclature, level of specification or structure of relationships. This means that although two ontologies describe the same domain of knowledge, at the computational level they are distinct structures, giving rise to a new problem of semantic heterogeneity.

To overcome this problem and take advantage of ontological modeling, the solution found is the so-called ontology matching (OM) process (EUZENAT; SHVAIKO, 2013). This process leads two heterogeneous ontologies into a mutual agreement by detecting a set of correspondences, called alignment. Ontology matching is a complex problem and its characteristics allow it to be approached by several computational techniques. Due to the high heterogeneity of the ontologies, there is no technique that stands out from the others in all aspects (XUE; TANG, 2017).

As many computational solutions can be used, comparing different approaches can be a challenging task due to the need for a comparison criterion. To coordinate and monitor the progress of OM research, the Ontology Alignment Evaluation Initiative¹ (OAEI) provides a set of datasets that can be used by researchers. OAEI also proposes an annual campaign to verify the quality of the solutions found by each tool. The OM problem is very wide, encompassing different types of semantic relations and other characteristics that hinder the development of a tool that is generic enough to obtain good results in all types of particularities, making researchers choose to work with sub-problems of the area. A sub-problem can be associated with the search

¹ <http://oaei.ontologymatching.org/>

for specific semantic relations, or with finding an algorithm that enhances the results of a processing step, for example. Researchers generally choose to use the OAEI dataset that best suits the sub-problem they aim to solve.

Ontology meta-matching (OMM) approaches are a specialization of ontology matching and have achieved good results in pairs of ontologies with different types of heterogeneities. OMM approaches seek to select the appropriate algorithms and their associated weights and thresholds (MARTINEZ-GIL; ALDANA-MONTES, 2012), in order to deliver an adaptive computational solution capable of achieving accurate results when facing different types of heterogeneities. In this scenario, it is common to find approaches based on evolutionary algorithms, which in recent years are appearing as the most suitable methodology to address the OMM problem (XUE; PAN, 2018).

Even with a well-defined workflow and consolidated datasets like those of OAEI, comparing OMM approaches can still be a challenging task, since it is necessary to define a set of algorithms, combination methods and other variables that will be used at each execution step of the workflow. As it is a process that requires several steps to build the final solution, the challenge is to identify which computational solution proposed by the researcher has made the greatest contribution in building an alignment and, in fact, represents research progress.

1.1 OBJECTIVES

There are several ways to address the problem, for example, through local optimization meta-heuristics (MOHAMMADI; HOFMAN; TAN, 2019), global optimization meta-heuristics (GULIĆ; VRDOLJAK; PTIČEK, 2018), supervised approaches (BINIZ; AYACHI, 2018), non-supervised approaches (XUE; LIU, 2017c), among other classifications. All of these types of approaches are capable of generating solutions to the problem. The main objective of this research is to investigate the impact that the choice of certain algorithms have on the quality of the results of a OMM approach. In this master thesis, it is first investigated how the workflow for processing OMM approaches can be interpreted and which are the most common algorithms. Then, a scientific experimentation framework is proposed and used to analyze costs, benefits, behaviors, and impacts that the algorithms cause on the final solution.

1.2 ONTOLOGY META-MATCHING

An ontology is a formal and explicit specification of a shared conceptualization (GRUBER, 1993a), where "formal" means machine readable, "explicit specification" refers to concepts, properties, relations, functions, restrictions, axioms, defined explicitly; "shared" means consensual knowledge; and conceptualization refers to an abstract model of some phenomenon of the reality. The matching task between two ontologies (source and target) consists of the search for semantic relationships (correspondences) between entities of source ontology with entities

of target ontology (ACAMPORA et al., 2013a). The term entity in this case is broader than it is in classic database models, i.e., the entity-relationship model; in an ontology all objects are first-class citizens, this includes classes, relationships, instances, data types, and values.

Definition 1. (Similarity measure). Let E_1 be the set of entities from o_1 and E_2 be the set of entities from o_2 , where $o_1, o_2 \in O$ and O is a set of ontologies. A similarity measure $sm : E_1 \times E_2 \rightarrow \mathbb{R}$ assesses the similarity between two entities $e_1 \in E_1$ and $e_2 \in E_2$ into a score $\eta_{1,2}$ where $\{\eta_{1,2} \in \mathbb{R} | 0 \leq \eta_{1,2} \leq 1\}$. An entity similarity measure is also called matcher.

There are different types of relationships. An entity $e_i \in E_1$ can be either equivalent, disjoint, or subsumes to another entity $e_j \in E_2$, for example. In the context of this master thesis, these relationships are limited to equivalence relationships.

Definition 2. (Alignment). The alignment is the output of the matching process. An ontology alignment A is a set of tuples (or correspondences) in the form $(e_i, e_j, \diamond_{i,j}, \eta_{i,j})$ where e_i and e_j represents two entities from distinct ontologies, followed by the relation descriptor \diamond representing a relationship type and $\eta_{i,j}$ is the confidence value of the correspondence.

Definition 3. (Ontology matching task). Let O be a set of ontologies and A a set of alignments. An ontology matching task $omt : O \times O \rightarrow A$ maps $o_1 \in O$ and $o_2 \in O$ to an alignment $a \in A$ through the use of a similarity measure.

The term meta-matching systems was presented by (EUZENAT; SHVAIKO et al., 2007) to describe systems that automatically parametrize a set of similarity measures (matchers). A meta-matcher approach deals with a set of candidate correspondences provided by several single matchers and is intended to combine the results and extract the most probable alignment.

Definition 4. (Ontology Meta-Matching task). Let SM be the set of similarity measures. A ontology meta-matcher $omm : SM \rightarrow A$ combines the similarity score of each $sm \in SM$ for each pair $\langle e_i, e_j \rangle$ where $e_i \in E_1$ and $e_j \in E_2$ into a unified score $\eta_{e_i, e_j} \in \mathbb{R}$ where $\{\eta_{e_i, e_j} \in \mathbb{R} | 0 \leq \eta_{e_i, e_j} \leq 1\}$ produces the best alignment $a \in A$.

Finding the best set of parameters to combine the similarity measures is a complex task and it is currently addressed through the use of meta-heuristics (XUE; CHEN, 2019b; SEMENOVA; KUREYCHIK, 2016b; SOUZA; SIQUEIRA; NUNES, 2019). Each meta-heuristic uses one or more objective functions to evaluate the quality of a solution. The solutions are proposed using a supervised (using a reference alignment) or an unsupervised approach. The main goal is to define the final confidence value for each candidate match through intelligent exploration of the search space. The final set of candidate correspondences is then subjected to a process to select the matches that have obtained the highest confidence rates. The set of selected correspondences is the answer to the problem; this set is called alignment. Some solutions,

however, are modeled in such a way that the correspondence selection process is unnecessary (XUE et al., 2018a; MOHAMMADI; HOFMAN; TAN, 2019).

To evaluate the result of the OMM algorithm, the most commonly used measures are Precision and Recall metrics (MANNING; RAGHAVAN; SCHÜTZE, 2008). These metrics compare the obtained alignment with the expected alignment. In some application scenarios such as the query answering, measures that assess processing time become more relevant to the experiment (SHVAIKO; EUZENAT, 2013).

The development of OMM approaches, as any other machine learning approach, requires a battery of experiments in order to refine the solution. Throughout these experiments, various informations regarding the execution of the process is generated and much of this information is lost at the end of the experiment. Much is discussed about how to make the results more reliable and how to make the experiment consolidated. According to (SHVAIKO; EUZENAT, 2013), for matching systems to gain broader acceptance and be trusted by users, they will need to provide explanations of their results to users or other programs that exploit them. Further discussion about solutions to provide explanations of OMM approaches and state-of-the-art of OMM techniques and evaluating costs/benefits of OMM algorithms will be included throughout this document's chapters.

1.3 CHAPTERS OVERVIEW

This master thesis is divided into five chapters. Chapters 2, 3, and 4 are full articles, while Chapter 5 connects the previous three chapters, explaining how each article is connected to the others, their findings, and the general context of the research. The results of this study were submitted to peer-reviewed conferences or journals, and each chapter of this dissertation is structured as manuscripts:

Chapter 2 Literature Review². To determine the set of elements that are part of the current OMM solutions, a systematic mapping of the literature was carried out. This chapter presents the result of this mapping, and its main objective is to understand the state of scientific research in OMM. This chapter has been submitted to the Journal Expert Systems with Applications and awaits feedback from reviewers.

Chapter 3 Proposed Approach³. A framework was designed to assist in the experimentation of OMM approaches. This chapter presents the architecture of the framework, as well as the available algorithms and other resources implemented to assist scientific experimentation in OMM. This chapter was published in Journal of Intelligent Information Systems and details can be found in (FERRANTI et al., 2020).

² Authors: Nicolas Ferranti, Jairo Francisco de Souza, Stênio Sã Rosário Furtado Soares

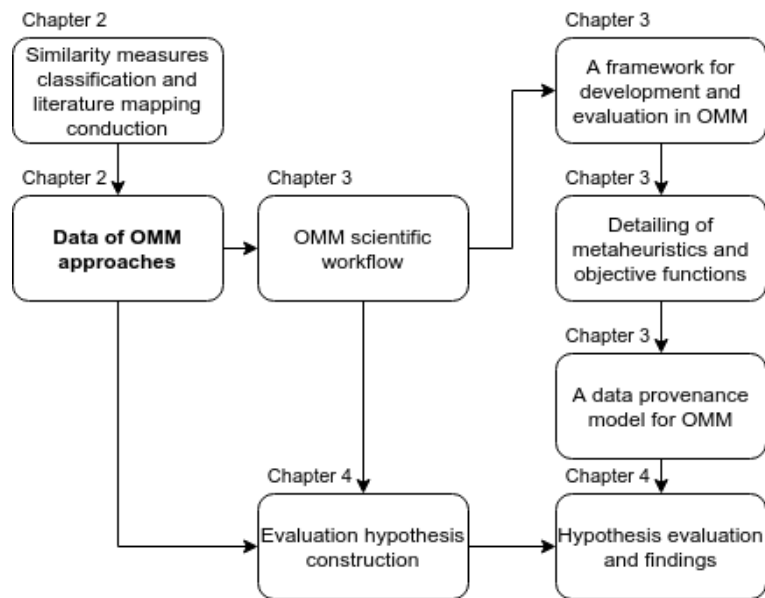
³ Authors: Nicolas Ferranti, José Ronaldo Mouro, Fabrício Martins Mendonça, Jairo Francisco de Souza, Stênio Sã Rosário Furtado Soares

Chapter 4 Experiments⁴. With the OMM experimentation framework completed, a series of algorithms pertaining to each stage of the workflow were evaluated. This chapter presents a comparative analysis of these algorithms and how they were able to answer a set of research questions.

Chapter 5 Conclusions and future works. Finally, this chapter discusses the methodologies and findings of each of the previous chapters, presenting the main contributions, limitations, and the perspectives for future work.

Figure 1 shows the workflow of the steps taken in the research conduction and in which Chapter the reader can find details of each step. Through Figure 1 readers can go through the document and find the subject that interests them most.

Figure 1 – Schematic of steps taken in the research conduction divided by chapters



Source: created by the author.

1.4 CONTRIBUTIONS

The main contributions of this work are presented below:

- A systematic mapping on metaheuristics-based OMM approaches covering seven years of published researches. The study presents a map of the implementation details usually used in unsupervised and supervised OMM approaches. To the best of your knowledge, it is the first systematic mapping on OMM.
- A modularized framework designed for scientific experimentation in OMM that already has the main algorithms used by researchers and that facilitates the process of implementing

⁴ Authors: Nicolas Ferranti, Jairo Francisco de Souza, Stênio Sã Rosário Furtado Soares

new approaches; The implementation is freely available for further research⁵. All the details of the framework can also be found in (FERRANTI et al., 2020).

- A novel prey-predator algorithm-based approach for OMM. The implementation is available in the framework and its results were detailed in (FERRANTI; SOARES; SOUZA, 2018b).
- A comparative case study among the main solutions applied in dealing with the problem of OMM; The results of this study pointed to a set of similarity measures that stand out from the others, as well as statements about the use of local optimization meta-heuristics, and the pros and cons of some semi-supervised and unsupervised objective functions.

⁵ <<https://bitbucket.org/nicolasferranti/heuristicontologymatching/src/master/>>

2 MANUSCRIPT 1: LITERATURE REVIEW

Ontologies have emerged to establish a well-defined meaning for information, solving problems of heterogeneity in data semantics and facilitating the process of information exchange. However, ontologies have generated a new semantic problem, since using more than one ontology can generate ambiguity in the meaning of a given data. The problem of ontology matching is to search for relationships between entities of distinct ontologies, solving the problem of semantic heterogeneity of the data. The problem is relatively new and several approaches have been proposed to solve it. In this master's thesis, the main objective is to evaluate costs and contributions of algorithms commonly applied to the problem. This chapter is a full article that presents a systematic mapping of the main works published in the area with an emphasis on metaheuristics-based meta-matching approaches. The main contribution of this chapter is the identification of the most common and promising computational solutions used in the ontology meta-matching literature.

2.1 INTRODUCTION

Over the years, technological evolution has been increasing the amount of information published by human and non-human agents on the web, forcing applications to evolve to be able to manage ever larger volumes of data. In addition to the problem of managing large amount of data, it is necessary to communicate with other applications in real time and to exchange fundamental information so that an application reaches its final goal. Main problem is heterogeneity in data representation (SHVAIKO; EUZENAT, 2013), which hinders access and creates ambiguity in the data semantics. Semantic Web (BERNERS-LEE et al., 2001) has brought the use of standardized descriptive models that help the process of information consumption. Ontologies are the main component of the Semantic Web to deal with the heterogeneity by specifying a semantic meaning for data. Yet, different goals lead ontology engineers to develop different ontologies that represent the same knowledge domain. In fact, the whole process of building an ontology is subjected to the engineers levels of expertise and domain view. Therefore, concepts that describe the same type of object may be represented in different ways, both in syntax terms and structure of relations, generating a problem of heterogeneity in data semantics.

Ontology matching problem consists of defining relationships between concepts of the input ontologies, making the structures compatible to represent the union of the datasets in a new model. Ontology matching is a complex problem and its characteristics allow it to be approached by several computational techniques. Due to the high heterogeneity of the ontologies, there is no technique that stands out from the others in all aspects (XUE; TANG, 2017).

Ontology meta-matching (OMM) is a subarea of ontology matching that aims to address the OM problem by combining different similarity measures. OMM approaches use methods to select the correct matching components to execute, and to adjust the multiple knobs (e.g.

threshold, coefficients, weights, etc.) of components that are part of the OMM process (XUE; PAN, 2018). In recent years, it is common to find approaches based on evolutionary metaheuristics, which have been presented as the most appropriate methodology to address the problem of OMM (XUE; PAN, 2018). This chapter provides a systematic mapping on the recent work in ontology meta-matching (OMM) area. The main contributions of this chapter are:

- A systematic mapping on metaheuristics-based OMM approaches covering published researches from 2013 to 2019;
- An analysis on implementation details usually used in unsupervised and supervised OMM approaches;
- Answering a number of research questions that can be useful for guiding new researchers in this field;

The remaining of this chapter is organized as follows. Section 2.2 formalizes the main concepts, while section 3 presents research methods. Section 4 describes data analysis of the papers found and section 5 contains the final considerations of this chapter.

2.2 ONTOLOGY MATCHING

An ontology is a formal specification of concepts and objects within a given domain and the relationships between them (GRUBER, 1993b; ACAMPORA et al., 2013a).

Usually, an ontology matcher returns a η value for each entity pair $\langle e_i, e_j \rangle$ that composes a candidate correspondence c . The η value represents the trust assigned to the relationship c , that is, how much it believes that the correspondence c is a match, where $\eta \in [0, 1]$. There is different types of correspondence. An entity e_i can be either equivalent, disjoint, or subsumes to another entity e_j , for example. A candidate match can be formalized as a tuple $(\langle e_i, e_j \rangle, \diamond, \eta)$, where e_i and e_j represents two entities from distinct ontologies, followed by the relation descriptor \diamond representing a correspondence type and η is the confidence value for that correspondence relation. A meta-matcher approach deals with a set of candidate correspondences provided by several single similarity measures (matchers) and is intended to combine the results and extract the most probable correspondences. The term meta-matching systems was presented by (EUZENAT; SHVAIKO et al., 2007) and describes systems that automatically parametrize a set of similarity measures.

Ontology similarity measures are classified considering two aspects (OTERO-CERDEIRA; RODRÍGUEZ-MARTÍNEZ; GÓMEZ-RODRÍGUEZ, 2015): based on the kind of input or the granularity/input representation. The description of each type is shown below and more details are given in Figure 2:

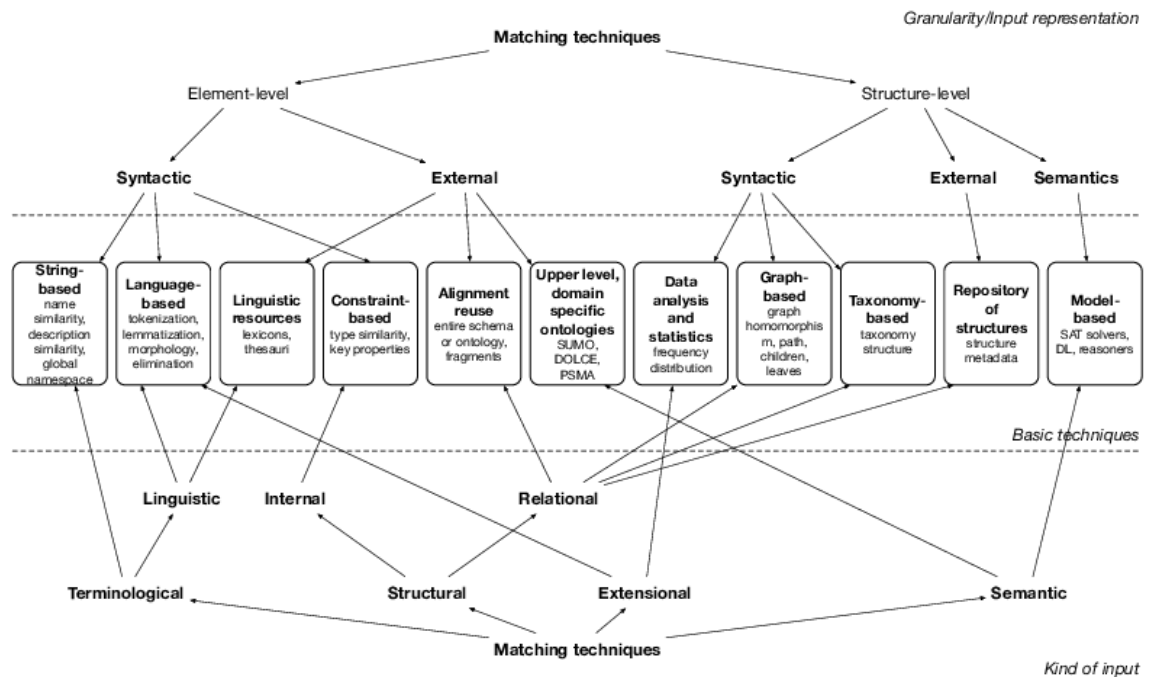
- **Ganularity/Input representation** : is based on how the techniques generally interpret the input information, how specific the matcher is to the point of dealing with the entire structure of ontology or just an attribute of an entity, i.e, element- or structure-level.
 - Element-level: techniques that obtain the correspondences by considering the entities in the ontologies in isolation, ignoring that they are parts of the structure of the ontology.
 - Structure-level: techniques that obtain the correspondences by analyzing how the entities fit within the structure of the ontologies.
- **Kind of Input** : is based on the kind of input which is used by elementary matching techniques. The first level of the hierarchy is described below:
 - Terminological : Terminological similarity measures are based on word similarity. There are several ways to verify the similarity between two words, in ontology matching there are three types of approaches that stand out: consider words (strings) as a sequence of letters in an alphabet and analyze the similarity between the way the word is written (string-based) or use an external knowledge base to verify the semantic field of words (Linguistic resources), use natural language processing techniques to evaluate the morphological properties of words (Language-based).
 - Structural : Structural techniques are based on the structure of the entities (classes, individuals, relations). They consider ontology entities to compare their relations with other entities. Generally use data type constraints or graph walking algorithms.
 - Extensional : Group of techniques based on data instances, highlighting category data analysis and statistics which represent techniques that take advantage of a sample of a population in order to find regularities and discrepancies.
 - Semantic : This class represents alignment at the semantic level, either through the use of other ontologies as background knowledge or through the use of reasoners that are commonly employed to find logical inconsistencies in an alignment.

Most meta-matching papers describe the matchers they use based on Figure 2, descending in the hierarchy shown to better specify the characteristics of the matcher. Full classification description is presented in (EUZENAT; SHVAIKO, 2013).

2.3 RELATED WORKS

Ontology matching research is relatively recent, but has a large scope of development. (OTERO-CERDEIRA; RODRÍGUEZ-MARTÍNEZ; GÓMEZ-RODRÍGUEZ, 2015) highlights the growth in the number of articles published annually between 2003 and 2012. New subareas have emerged due to the need to solving increasingly complex problems, such as large-scale

Figure 2 – Classification of the similarity measures



Source: (EUZENAT; SHVAIKO, 2013)

ontology matching, complex matching and ontology meta-matching. Although there are known reviews on ontology matching (OTERO-CERDEIRA; RODRÍGUEZ-MARTÍNEZ; GÓMEZ-RODRÍGUEZ, 2015; SHVAIKO; EUZENAT, 2013), few published papers shed light on evolutionary meta-matching approaches. Sections 2.3.1 and 2.3.2 discuss how ontology matching and ontology meta-matching techniques, respectively, have been compared in literature reviews.

2.3.1 Ontology matching reviews

The main review on ontology matching is based on evidence-based software engineering, presenting a formal and reproducible methodology. The selection of articles usually takes place through classic systematic review processes or using the Ontology Alignment Evaluation Initiative¹ (OAEI) as a reference. In the first case, a search string is defined for execution in online article repositories, then the results are filtered until only relevant articles remain according to the authors' criteria. In the second case, works submitted to OAEI campaigns are selected. OAEI is the leading organization for evaluating ontology matching approaches and it provides test dataset for several alignment problems. The approaches submitted to the OAEI campaigns are a good indication of the state of research in ontology matching. The work of (OTERO-CERDEIRA; RODRÍGUEZ-MARTÍNEZ; GÓMEZ-RODRÍGUEZ, 2015) combines the two cited article selection processes, while the works of (SHVAIKO; EUZENAT, 2013; MOHAMMADI; HOFMAN; TAN, 2018) discuss works submitted to OAEI campaigns only.

¹ <http://oaei.ontologymatching.org/>

Regarding data analysis, similarity measures are often used in review papers to compare ontology matching systems. Similarity measures analyze the similarity between entities in different ways. (OTERO-CERDEIRA; RODRÍGUEZ-MARTÍNEZ; GÓMEZ-RODRÍGUEZ, 2015) presents a classification system for similarity measures, exemplifying with papers that use such a measure. In (SHVAIKO; EUZENAT, 2013; ARDJANI; BOUCHIHA; MALKI, 2015) the similarity measures used in each approach are also compared. In turn, (ARDJANI; BOUCHIHA; MALKI, 2015) counts how many works a group of measures is used for.

Throughout the review analysis, some authors foresee the future of the area. For instance, (SHVAIKO; EUZENAT, 2013) discusses eight challenges for research evolution, such as user interaction, explanation of results, and the effectiveness of similarity measures. Overall, (SHVAIKO; EUZENAT, 2013) suggest that an ontology matching approach should deliver more than just the correspondences set, bringing more robust analysis and bringing the matching tools closer to end users. In its turn, (OTERO-CERDEIRA; RODRÍGUEZ-MARTÍNEZ; GÓMEZ-RODRÍGUEZ, 2015) highlights the small number of real-world applications of ontology matching approaches when comparing to the number of published articles, and also discuss future challenges, such as scalability issues and automation of the approaches.

The variety of computational techniques that can be used in ontology matching makes it difficult to review papers when it comes to evaluating the matching method of each paper. There are works ranging from artificial neural networks (DJEDDI; KHADIR, 2013) to heuristics (NGO; BELLAHSENE, 2016) or metaheuristics approaches (XUE; TANG, 2017; ACAMPORA et al., 2013a; SINGH, 2018). This difficulty in evaluating search-based methods makes works such as (OTERO-CERDEIRA; RODRÍGUEZ-MARTÍNEZ; GÓMEZ-RODRÍGUEZ, 2015; SHVAIKO; EUZENAT, 2013; RANA; SINGH, 2014) analyze the functioning of each approach separately. To specialize a review in an ontology matching subarea, such as evolutionary meta-matching, can facilitate the comparison of each approach.

2.3.2 Ontology meta-matching reviews

Ontology meta-matching is related to the process of selecting the most appropriate algorithms, matchers and weights in various matching scenarios in order to achieve quality alignment. The process of meta-matching usually occurs through optimization algorithms, where evolutionary metaheuristics are often employed due to their exploratory capability over search spaces and because of the complexity of the matching problem (XUE; CHEN, 2018).

Most studies comparing meta-matching approaches do not follow a formal article selection methodology. The studies usually compare metaheuristics implemented in the same environment within the same tool (ACAMPORA; ISHIBUCHI; VITIELLO, 2014; ACAMPORA; VITIELLO, 2020). For instance, (ACAMPORA; ISHIBUCHI; VITIELLO, 2014) focuses on evolutionary multi-objective approaches, further limiting analysis to multi-objective approaches. The authors aimed to evaluate the optimization step, where a meta-heuristic was

used to combine similarity measures. This analysis is more quantitative than qualitative, making use of hypervolume, δ index and C metrics. It is interesting to highlight the use of these metrics, as researchers usually evaluate their approaches using only precision, recall, and f-measure values, leaving aside other aspects that may be relevant. The main finding in (ACAMPORA; ISHIBUCHI; VITIELLO, 2014) is the statistical performance of the Optimal Multi-Objective Particle Swarm Optimization (OMOPSO) over other famous multi-objective algorithms such as NSGA-II, MOEA/D, and SPEA2. The most recent work in the area (ACAMPORA; VITIELLO, 2020) focuses on the performance of local search metaheuristics and highlights the TABU search performance.

Related works demonstrate the growth of the ontology matching area and the efforts of researchers to summarize the main findings so far, especially in the last decade. While few reviews on ontology matching were made using systematic methods for selection and comparison of articles (OTERO-CERDEIRA; RODRÍGUEZ-MARTÍNEZ; GÓMEZ-RODRÍGUEZ, 2015), the evolutionary meta-matching subarea still lacks a literature review using this systematic method.

2.4 RESEARCH PROTOCOL

In this work a systematic review of evolutionary ontology meta-matching is performed, using reproducible methods to select and compare the main articles published since 2013, the year in which book (EUZENAT; SHVAIKO, 2013) was published, this book summarizes the main efforts and advances of OM researchers. The objective of this study is to identify which techniques are most used for ontology meta-matching. The majority of approaches models the ontology meta-matching as a combinatorial optimization problem and uses meta-heuristics to find solutions in a feasible processing time. During the planning activity it was identified the objectives and defined the protocol. The protocol specifies the method to be used in the systematic mapping in order to reduce researcher bias (STEINMACHER; CHAVES; GEROSA, 2013) and make the process reproducible. Considering the predominance of evolutionary approaches and guided by (PETERSEN et al., 2008), we propose the following research questions:

- RQ1: How many studies were published over the years?
- RQ2: Who are the most active authors in the area?
- RQ3: Which publication vehicles are the main targets for research production in the area?
- RQ4: What are the most common similarity measures used?
- RQ5: What are the main optimization meta-heuristics used?
- RQ6: How have solutions been represented?

- RQ7: What are the main objective functions?
- RQ8: Which datasets have been used for OMM evaluation?

Based on the research questions, the PICO (Population, Intervention, Comparison, and Outcomes) (KEELE et al., 2007) method was used to develop a search scope presented in detail in Table 1. The comparison entry is not relevant in this work, since this mapping aims to provide an overview of the subject through an exploratory study.

Table 1 – PICO Elements

Element	Scope
Population (P)	Ontology Matching, more specifically Ontology meta-matching
Intervention (I)	Approaches for ontology meta-matching
Comparison (C)	No comparison intervention
Outcomes (O)	Computational solutions

Source: (KEELE et al., 2007)

The main terms were extracted from the research questions and combined with those from PICO to create the search string. In order for expanding the query, the search string contains synonyms for PICO terms (Table 2). The string is composed of the previously mentioned terms concatenated by AND/OR operators, where the OR operator groups terms from the same PICO element. The AND operator groups sets of terms from different PICO elements to ensure that at least one of each occurs in the articles. This work also used control articles to evaluate the search string, which means that a previous set of articles characterized as primary studies were used to assist in the creation of the string, acting as expected articles.

Table 2 – PICO Keywords

Element	Keywords
Population	“ontology matching”, “ontology meta-matching”
Intervention	“evolutionary”, “meta-heuristic”, “metaheuristic”, “multi-objective”
Outcomes	“approach”, “method”, “solution”, “algorithm”

Source: created by the author.

The final search string was defined as: (*"ontology matching"OR "ontology meta-matching"*) AND (*"evolutionary"OR "meta-heuristic"OR "metaheuristic"OR "multi objective"*) AND (*"method"OR "approach"OR "solution"OR "algorithm"*).

Digital repositories for search execution were selected based on three criteria: accept boolean expressions, cover Computer Science domain and are available at the researcher's

institution. The following repositories were selected: IEEE Xplore Digital Library², Scopus³, Google Scholar⁴.

Both IEEE Xplore and Scopus have tools to refine the search, thus the search string was executed on the title, keywords and abstract of the publications. The advanced search of Google Scholar does not allow this specific filtering, then the search was performed on the entire publication text. After performing the search, the process used to include and exclude a paper was organized in the following exclusion (EC) and inclusion (IC) criteria:

- (EC1): Article was published before 2013 OR
- (EC2): Article is not written in English language OR
- (EC3): Article is gray literature OR
- (EC4): Article full text is unavailable
- (IC1): Article discusses the ontology meta-matching problem AND
- (IC2): Article is written in English language AND
- (IC3): Article presents a proposal of a metaheuristic-based solution for the ontology meta-matching optimization step.

The articles were analyzed on six characteristics, as follows:

- *C1*: Similarity measures. Papers must clearly state what similarity measures were used.
- *C2*: Supervision. Papers should specify if they use reference correspondence sets for training, that is, if the process is supervised or unsupervised. This information could also be inferred from the description of the objective function.
- *C3*: Objective Functions. Papers should clearly specify the objective function.
- *C4*: Objectives. It is a single or multi-objective approach?
- *C5*: Optimization Meta-heuristics. Papers should clearly specify which metaheuristic was used.
- *C6*: Solution Representation. Papers should clearly specify how solutions (individuals) are represented in the proposal.

² <https://ieeexplore.ieee.org/Xplore/home.jsp>

³ <https://www.scopus.com/home.uri>

⁴ <https://scholar.google.com.br/>

For a paper to be present in the final set, it is necessary to present at least four of the six characteristics (quality assessment step). Finally, the article selection process took place in steps that are described as follows: (1) Search string execution in repositories, (2) Duplicates removal, and (3) Full text reading. The final list of publications were analyzed according to their characteristics and the information was extracted and stored. Section 2.5 presents the analysis of these data. Quantitative data are mostly presented in graphs while the qualitative analysis is discussed throughout the section.

2.5 MAPPING REPORT

The search was performed in August 2019 and returned 1137 publications. Since Google Scholar has no filter to only title, keywords and abstract, is predictable that it will return more publications than the others. Table 3 shows the retrieved papers per repository.

Table 3 – Amount of retrieved papers from each repository

Digital Library	Amount of Papers
Google Scholar	1080
Scopus	49
IEEE Digital Library	8

Source: created by the author.

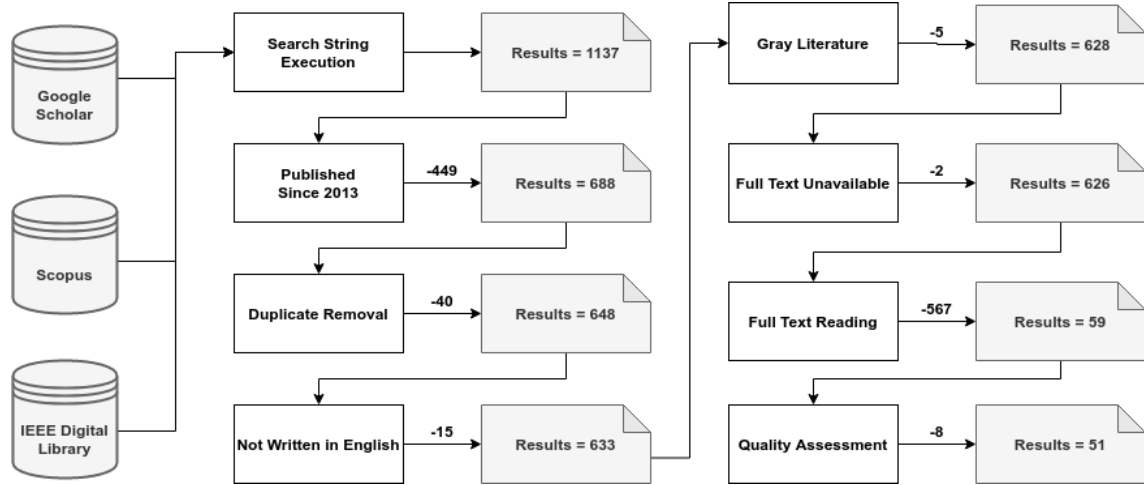
In the first filter step, 449 articles were removed by EC1, reducing the total amount to 688. In **Duplicate Removal** step, 40 articles were removed. Based on EC2, 13 publications were removed, followed by 5 on EC3 and 2 on EC4. After full text reading, 567 were removed. Finally, 8 articles were excluded because they did not meet the minimum quality according to their characteristics. In the end, 51 were selected to answer the research questions. The remaining articles represent approximately 4% of the total. Figure 3 details each step of the mapping conduction. All data is freely available for future verification and analysis in our online repository⁵. The following subsections discuss each one of the research questions.

2.5.1 RQ1: How many studies were published over the years?

Figure 4 represents the answer to RQ1 graphically. In this systematic mapping, the range of years was limited to works published from 2013 onwards to observe publications not captured by (OTERO-CERDEIRA; RODRÍGUEZ-MARTÍNEZ; GÓMEZ-RODRÍGUEZ, 2015). (OTERO-CERDEIRA; RODRÍGUEZ-MARTÍNEZ; GÓMEZ-RODRÍGUEZ, 2015) demonstrates a decrease in the number of publications in 2013. Despite the large volume of articles found by (OTERO-CERDEIRA; RODRÍGUEZ-MARTÍNEZ; GÓMEZ-RODRÍGUEZ, 2015) compared to this work, our systematic mapping points to an increase in the volume of

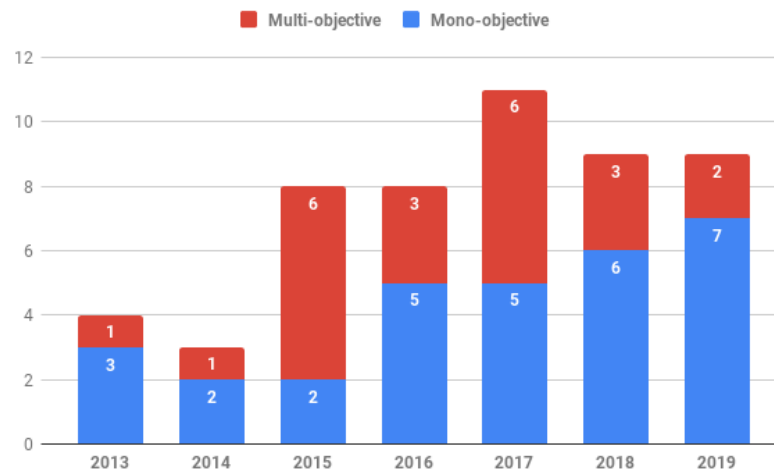
⁵ <https://github.com/lapic-ufjf/OMM-systematic-mapping>

Figure 3 – Systematic mapping conduction



Source: created by the author.

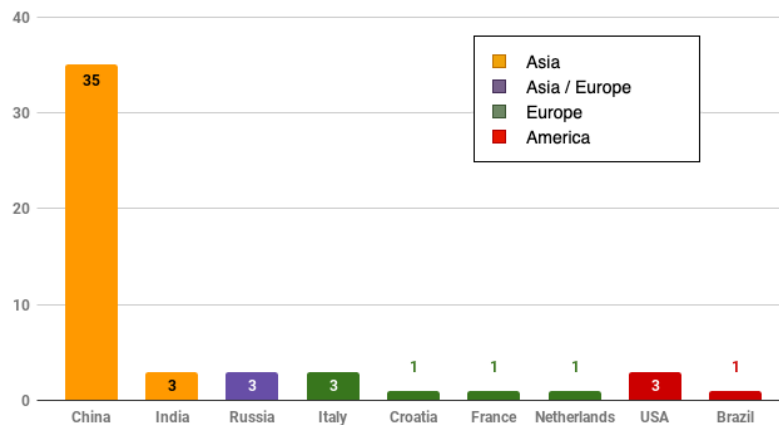
Figure 4 – Number of mono-objective and multi-objective approaches among the articles found by the systematic mapping



Source: created by the author.

evolutionary meta-matching approaches publications, indicating that the area has been growing over the last few years. Until mid 2019, 9 articles were published with evolutionary proposals for ontology meta-matching, equaling the total found in the previous year. The growth of evolutionary approaches in recent years may be related to the multi-objective approaches, multi-objective approaches have been gaining ground in recent years as shown in Figure 4 due to the possibility of including multiple aspects of the problem in the evaluation, improving the quality of the results. Some aspects that may be of interest to the researcher are: the size of the resulting alignment, the method of combining similarity measures, and the threshold to establish the minimum confidence for a candidate correspondence.

Figure 5 – Total articles found in the systematic mapping divided by country



Source: created by the author.

2.5.2 RQ2: Who are the most active authors in the area?

To answer RQ2, first table 4 is presented, showing the authors individually separated and ranked by the amount of works present in this mapping. Xingsi Xue is the most outstanding author with a total of 28 publications. Figure 5 distributes the publications by countries, pointing out that China is the country with the most publications in the area. Figure 5 was developed based on the institutional bond of the lead author.

Table 4 – Authors' names and number of publications

Name	Total
Xingsi Xue	28
Junfeng Chen	6
Jianhua Liu, Pei-Wei Tsai and Yuping Wang	4
Viktor Kureychik, Alexandra Semenova, Giovanni Acampora, Vincenzo Loia, Jinshui Wang and Aihong Ren	3

Source: created by the author.

In order to analyze the contributions among the authors, a study on the citation network was performed. This type of study also helps to highlight the most relevant publications in the community, these publications usually stand out for the high number of citations. The result of this process is shown in the citations graph (Figure 6).

After filtering only citations to the mapped articles, 49 articles were found. According to Fig 7, the citation graph can be read as follows: each sector (E) of the graph represents a specific article identified by a number (A) in the range [1, 49] (Table 27 in the appendix maps each identifier to its respective paper). The colors used in each sector indicate the year in which the paper was published. Between each distinct paper (sector), there is a set of rows. Each line

starts on an inner circle (F) and ends on an outer circle (G). The outer circle represents the paper being cited and the inner circle represents the paper that cited. The colourful edges represent the publishing year of the paper in which the citation was made. Around the sectors are two to three arcs, each with an array of colors. They represent, from the inside out, the outbound citations (C), the inbound citations (D), and the sum of them all (B). The more distinct the colors, the more the specific article interacted with others. The graph is better viewed with larger size ⁶

Article number 1 (published in 2013) and article number 20 (published in 2015) are the most cited in the network of mapped articles, totaling 15 and 19 citations respectively, this data demonstrates the importance of these two publications for the area, mainly among the set of publications that took place from 2017 onwards. Paper number 1 (ACAMPORA; LOIA; VITIELLO, 2013), is the first of this mapping to use a memetic approach to compute the weights of matchers, this article stands out for its use of a mono-objective fitness function that combines two goals through a predefined weight, giving evidence that the area could explore multi-objective approaches. Another highlight is the accuracy, which is around 94%, a high value given that the (ACAMPORA; LOIA; VITIELLO, 2013) approach is unsupervised. Article number 20 (XUE; WANG, 2015b) also presents a memetic approach, and the objective function simulates precision and recall measurements without the need for a reference alignment, making this approach unsupervised as well. (XUE; WANG, 2015b) is cited mainly by articles from 2018 and 2019, most of these citations come from the same main author, indicating the evolution of the research.

2.5.3 RQ3: Which publication vehicles are the main targets for research production in the area?

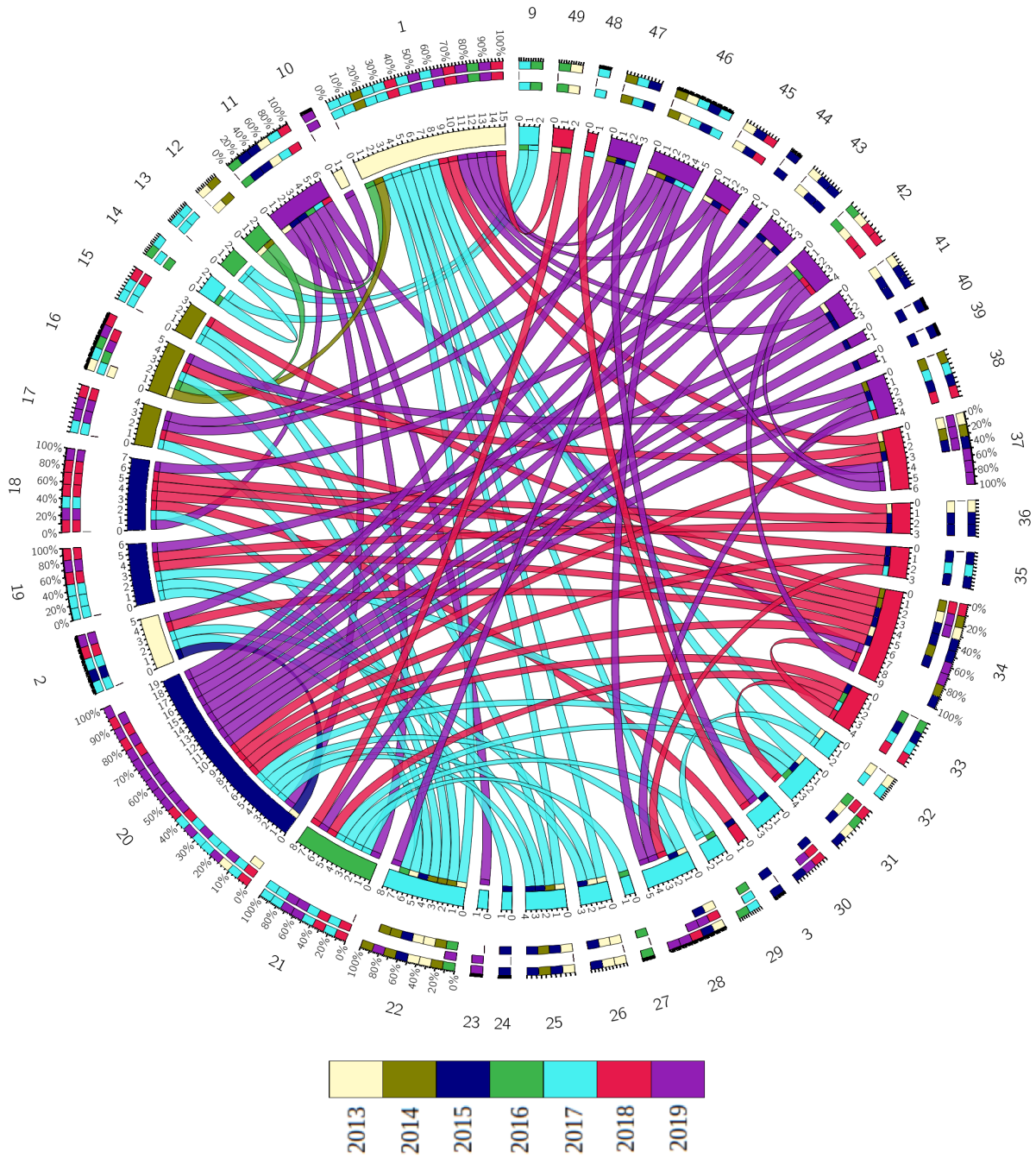
Table 5 shows where the selected papers were published, the vast majority (approximately 69%) of the papers were published in journals. In order to identify which journals are most popular among the selected papers, Table 6 is presented also containing the h-index metric value.

Journal of Information Hiding and Multimedia Signal Processing is the most popular journal, however it represents only 14% of the journals total, this shows that although journals are more popular than conferences, there is no journal that concentrates a large subset of mapped articles, the articles are distributed in several distinct journals.

Conferences are valued as a destination for reporting research on Computer Science (CS) (MEYER et al., 2009; PATTERSON; SNYDER; ULLMAN, 1999). Unlike other academic fields, the length of conference papers in CS enables sufficient detail of the work to be reported and discussed. Although some researchers criticize the use of conferences as main dissemination route in CS (FORTNOW et al., 2009; VARDI, 2010), some authors argue that high quality conferences have a higher average citation rate compared to journals in the same stratum (VRETTAS; SANDERSON, 2015), and seem to serve as a distinct channel of academic

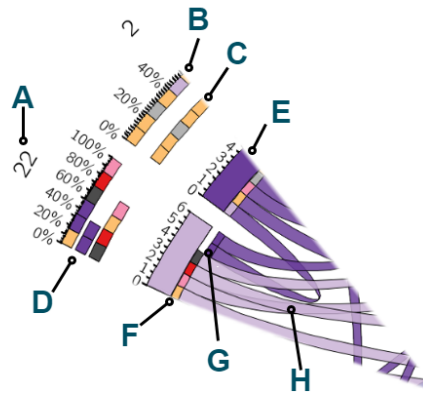
⁶ https://github.com/lapic-ufjf/OMM-systematic-mapping/blob/master/citations_chart-large.png

Figure 6 – Citations graph between the articles of the systematic mapping



Source: created by the author.

Figure 7 – Citations graph explanation



Source: created by the author.

communication, not as a mere step prior to journal publication (KIM, 2019). Results in Table 5 shows that conferences account for 31% of total publishing destinations, proving to be an important means of disseminating research in the area even though they do not outnumber journals. Table 7 shows that just as in journals, conference publications are also spread across several conferences, the most frequent representing only 19% of total conferences. Tables 6 and 7 reflect the interdisciplinarity of the problem, showing publication venues that are not necessarily from the database area.

Table 5 – Publishing Vehicle Types

Venue	Total
Journals	35
Conferences	16

Source: created by the author.

Table 6 – Most Popular Journals

Name	Total	H-index
Journal of Information Hiding and Multimedia Signal Processing	5	25
Knowledge and Information Systems	2	56
Information Sciences	2	154

Source: created by the author.

2.5.4 RQ4: What are the most common similarity measures used?

The classification presented in Section 2.2 was used to answer RQ4. The similarity measures were grouped based on the leaf nodes of Figure 2 so that more details can be explored. Table 8 shows the number of times that a type of similarity measure was found among the analyzed articles.

Table 7 – Most Popular Conferences

Name	Total
Intl. Conf. on Intelligent Information Hiding and Multimedia Signal Processing	3
Intl. Conf. on Computational Intelligence and Security	2
Intl. Conf. on Genetic and Evolutionary Computing	2

Source: created by the author.

Table 8 – Most commonly used similarity measures grouped by technique classification

Technique Classification	Total
String-based	61
Linguistic resources	40
Taxonomy-based	32
Language-based	6
Constraint-based	6
Data analysis and statistics	4
Model-based	1
Graph-based	1

Source: created by the author.

It was found eight types of similarity metrics. However, some metrics could be classified in more than one type. To better understanding the results in Table 8, the rest of the section provides details of each type and discusses the rationale for the classification performed.

String-based measures appear at the top of the list due to their low computational cost. They are usually based in the principle that the more similar the strings, the more likely they are to denote the same concepts. Among the articles that provided their similarity measures, 93% of the authors used at least one string-based similarity measure. Since no similarity measure stands out from the rest for any application scenario (XUE; LU; CHEN, 2019), a combination of similarity measures naturally occurs. This combination often occurs with measures from different groups to explore different aspects of ontology. The most commonly found string-based similarity measures were: Jaro-Winkler (WINKLER, 1999), SMOA (STOILLOS; STAMOU; KOLLIAS, 2005), and Levenstein distance (MAEDCHE; STAAB, 2002). Depending on the alignment domain, some string-based measures perform better than others. (XUE; HANG; TANG, 2019) state that N-gram-based measurement performs well for biomedicine ontologies while, according to (STOILLOS; STAMOU; KOLLIAS, 2005), the SMOA distance is shown to be the most performing string-based measure for the ontology matching problem when there is no prior knowledge for the alignment task.

Linguistic resources are approaches that use linguistic relationships between words, such as synonyms or hyponyms, to identify the similarity. These relationships are obtained through lexicons or domain-specific thesaurus. Linguistic techniques occupy the second position among the most found in the papers, 89% of the works use at least one linguistic-based measure. In this

category, the most common similarity measure uses WordNet for searching relationships (XUE; PAN, 2018; ACAMPORA et al., 2013a; BINIZ; AYACHI, 2018), probably due to the fact that WordNet is a generic lexical database, representing domain-free linguistic relationships. Some domain-specific ontology matching approaches use specific linguistic resources, such as the Unified Medical Language System (UMLS), which is often used to support matching between health ontologies (XUE et al., 2018c; XUE; LU; CHEN, 2019).

Taxonomy-based techniques act at a structural level, comparing the entities or instances of an ontology with other entities or instances. Taxonomic techniques are graph-based algorithms that consider only specialization relations. These techniques were used in 69% of the articles. The most commonly used technique is the similarity flooding algorithm (XUE; PAN, 2018; BINIZ; AYACHI, 2018; ZHANG; GUO, 2019) that uses a base match to propagate similarity to nearby nodes. Graph-based techniques are also structural techniques, however they use other relations, such as object properties. In these techniques, ontologies are considered as labeled graphs and similarity is computed based on the node position in the graph (ZHANG; GUO, 2019).

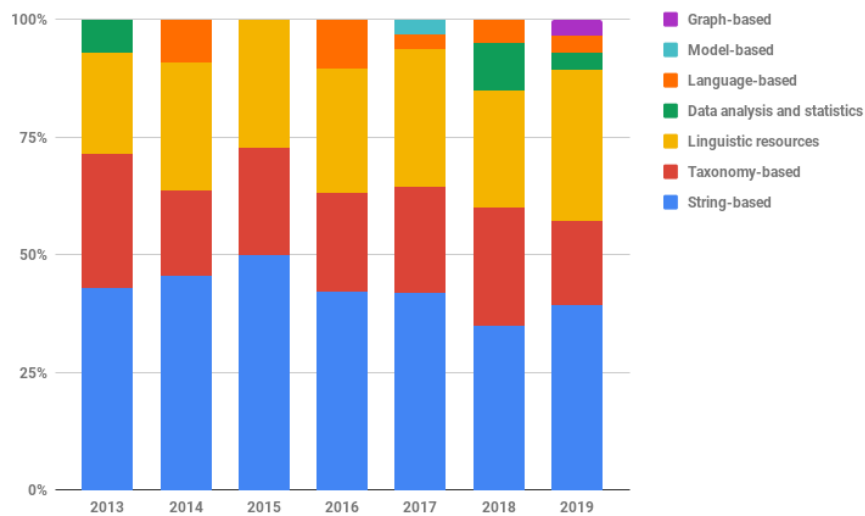
Language and constraint-based algorithms were used in six works. The first uses information retrieval techniques to process terms and compute the similarity (XUE; WANG; REN, 2016; MOHAMMADI; HOFMAN; TAN, 2019; KESSENTINI et al., 2014). The second is based on the domain and range restrictions of the entity properties (ACAMPORA et al., 2013a; ACAMPORA et al., 2013b). Data analysis and statistics are techniques that work with extensional data, usually instances, exploring a representative sample of a population in order to find regularities and discrepancies (BINIZ; AYACHI, 2018; XUE; CHEN, 2019b). Finally, model-based approaches manipulate the input based on its semantic interpretation. Model-based approaches are usually used at the end of the matching process to find logical inconsistencies in the returned alignment. This can be performed with the help of a reasoner, as done by (XUE; TANG, 2017).

Figure 8 shows the percentage of use of each type of technique over the last years. It is possible to observe the constant and representative participation of the string-based, taxonomy-based and linguistic resources measures. Approximately 57% of articles use at least one measure of each of the three types within their measurement set. Some studies reached high accuracy values, around 90% using these three types of measures, such as (XUE; LIU, 2017c). It is noteworthy that the approach presented in (XUE; LIU, 2017c) is unsupervised, and these measures can be used together to find the most varied types of correspondences.

2.5.5 RQ5: What are the main optimization meta-heuristics used?

In all 51 articles analyzed by this study, it was possible to identify which metaheuristics the authors made use of. In total 52 meta-heuristics were analyzed (two approaches are presented in (KESSENTINI et al., 2014)). Table 9 presents the most frequent metaheuristics grouped according to the metaheuristic classification, including articles in which the metaheuristic was

Figure 8 – Percentage of each type of similarity measure per year



Source: created by the author.

observed. Approximately 79% of the articles use genetic or memetic algorithms, indicating the good performance of these techniques over the last years but also a possible saturation of the literature. In the next subsections each metaheuristic is briefly described.

Genetic Algorithm

Genetic Algorithm (GA) (TALBI, 2009) was proposed by J. H. Holland in 1975 as a search technique inspired by the Darwin's Theory Of Evolution, in which "fitter individuals will have a higher probability to survive and pass their genes to the next generation through genetic operations". In a genetic algorithm, a population of candidate solutions is submitted to a set of processes that simulate the evolution process. The first process selects the individuals for reproduction according to the individual's performance on an objective function. In the next process the genes of the selected individuals are combined to generate new individuals. Offspring generated from the combination of parental genotypes are subject to mutation process, which causes characteristics not present in the population to be incorporated, functioning as a strategy of diversification in the search process. Eventually the replacement process creates the new population containing the fittest individuals. Throughout the process it is common for the population size to remain the same.

The evolutionary character of the algorithm lies in the fact that in each generation t (iteration of the algorithm) the population P_t contributes to the formation of the next generation population P_{t+1} from the operators (processes) described. The algorithm stops when reaches a given number of generations or converges to an established value. The solution is the best individual obtained throughout the evolutionary process. This whole process is considered computationally costly by many authors. In order to reduce the computational cost, some authors

Table 9 – Metaheuristics used in OMM literature

Metaheuristic	Total	Articles
Genetic Algorithm	27	(MARTÍNEZ-ROMERO et al., 2013), (XUE et al., 2015), (XUE; TSAI; ZHANG, 2016), (KESSENTINI et al., 2014), (XUE; TANG, 2017), (XUE; CHEN, 2019a), (GULIĆ; VRDOLJAK; PTIČEK, 2018), (XUE; WANG; REN, 2016), (XUE; PAN, 2018), (XUE; WANG; HAO, 2014), (XUE; LIU, 2017c), (XUE; TSAI; FENG, 2017), (XUE; CHEN, 2018), (ACAMPORA et al., 2013a), (XUE; WANG; HAO, 2015), (XUE; WANG, 2015a), (JIANG; XUE, 2015), (XUE et al., 2015), (XUE; WANG, 2017), (XUE, 2017), (XUE; REN, 2017), (XUE et al., 2018a), (BINIZ; AYACHI, 2018), (XUE; LU; CHEN, 2019), (JIANG et al., 2016), (XUE et al., 2018b), (XUE; LIU, 2018)
Memetic Algorithm	14	(XUE; LIU, 2017a), (XUE; CHEN; YAO, 2018), (XUE; WANG, 2015b), (XUE; WANG; REN, 2014), (ACAMPORA et al., 2013b), (XUE; TSAI; WANG, 2017), (ACAMPORA; LOIA; VITIELLO, 2013), (XUE; CHEN, 2019b), (XUE; CHEN; CHEN, 2019), (XUE; PAN, 2017), (XUE; LIU, 2017b), (RAMESH et al., 2016) (XUE; CHEN, 2019c), (XUE; HANG; TANG, 2019)
Particle Swarm Optimization	5	(KUREYCHIK; SEMENOVA, 2017), (SEMENOVA; KUREYCHIK, 2016b), (XUE; LIU; WANG, 2017), (MARJIT, 2015), (SEMENOVA; KUREYCHIK, 2016a)
Simulated Annealing	3	(MOHAMMADI; HOFMAN; TAN, 2019), (KESSENTINI et al., 2014), (XUE et al., 2018c)
Others	3	(ZHANG; GUO, 2019) , (FERRANTI; SOARES; SOUZA, 2018a) , (FORSATI; SHAMSFARD, 2016)

Source: created by the author.

replace the classical model of population representation with a probability vector (PV). PV is used to generate individuals who are used to update the PV throughout the evolutionary process.

Approximately 52% of the metaheuristics found in the articles are GAs, classic approaches are also present, in (XUE; TANG, 2017) selection is based on a queue where individuals are ordered by fitness, crossover uses the one-cut-point method, all classic GA operators.

Genetic algorithms have several variations, some ontology meta-matching researches make use of more than one objective function, where one function usually conflicts with the others, causing an adaptation of the algorithm to search for non-dominated solutions according to the Pareto curve. In this scenario the algorithm that stands out in the meta-matching problem is the NSGAI algorithm, approximately 44% of all GA publications has used a NSGA variant (ACAMPORA et al., 2013a; XUE; WANG, 2015a), making NSGA the most common of GAs. Another variation of GA found in smaller quantity is the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D). MOEA/D, which is also an algorithm for finding non-

dominated solutions, tackles the problem by breaking it into small sub-problems and creating islands with separate populations, each island is associated with a different objective function. MOEA/D uses the neighborhood concept to generate new individuals based on parents who may be from different islands, exchanging information between different populations. MOEA/D represents 11% of GA approaches.

Memetic Algorithm

Memetic algorithms are a class of optimization algorithms composed of Genetic algorithms that performs in each generation one or more local search components (XUE; LIU, 2017b). The combined use of global search and local search is helpful to reduce the possibility of the premature convergence and increasing the convergence speed (GARG, 2010). Generally, the local search process is applied in the best solution with each generation of the evolutionary algorithm.

Also called hybrids by many authors, memetic algorithms have been successfully applied in solving several combinatorial problems, making it very suitable for dealing with a complex task like the ontology matching problem (XUE; PAN, 2017). Approximately 27% of the approaches found are memetic algorithms.

Local search, which is the main differentiator of this approach for a GA, can be performed in a number of ways. (XUE; CHEN, 2019c) and (XUE; HANG; TANG, 2019) use the TABU search algorithm in the best solution for each evolutionary cycle while (XUE; TSAI; WANG, 2017). The conclusion is that the local search method is chosen according to the characteristics of the author's approach, such as the representation of the solution, time taken to process the local search and other factors.

Regarding similarity to the processes of a GA, some authors have further optimized hybrid algorithms by replacing the traditional GA population with a probability vector (PV) that characterize the entire population (XUE; CHEN, 2019b; XUE; CHEN; CHEN, 2019), as it was done in GAs. Classical evolutionary operators such as mutation and recombination are used over PV-generated individuals. The PV is updated throughout the evolutionary process (XUE; CHEN, 2019b). This optimization is interesting in dealing with the meta-matching of large scale ontologies (XUE; LIU, 2017b).

Particle Swarm Optimization

Particle swarm optimization (PSO) is a natural-inspired technique based on the social behavior of individuals, such as the flock of birds to find a place with food enough (SHI; EBERHART, 1998). Several factors contribute to the use of PSO in the OMM scenario: the performance in terms of convergence speed and ability to exceed local minimums equals the performance of GA (BOCK; HETTENHAUSEN, 2012). PSO convergence outperforms GA in several optimization problems because of the result of social information sharing, while GA is

based on competition between individuals (YU; XIONG; WU, 2004). Approximately 10% of the metaheuristics used are based on PSO, where both mono-objective (SEMENOVA; KUREYCHIK, 2016a) and multi-objective (MARJIT, 2015) approaches were found. In (SEMENOVA; KUREYCHIK, 2016a), the authors decided to choose PSO because this metaheuristic needs fewer parameters to adjust than other metaheuristics, such as GA. For the authors, the time taken by other metaheuristics to find good solutions can be high, and since PSO does not use complicated evolutionary operators like GA, it may be possible for the PSO approach to be faster than GA.

Simulated Annealing

Simulated annealing (SA) is a probabilistic metaheuristic to estimate the global optimum of a function (METROPOLIS et al., 1953). As its name implies, this metaheuristic is inspired by a thermal process of metallurgy, where low energy states are obtained in a solid. Unlike other metaheuristics presented so far, SA is not a population-based metaheuristic, some authors take advantage of this feature to justify the use of SA in memory gain and processing time when compared to GA and PSO (MOHAMMADI; HOFMAN; TAN, 2019). The approach proposed by (XUE et al., 2018c) explores the time factor of SA by applying this metaheuristic to the alignment of large-scale ontologies. A feature of SA is that it allows, more often at the beginning of the search process, to accept unimproved solutions to escape from local optimum solutions, this feature tends to avoid premature convergence.

In the area of OMM, SA represents approximately 6% of the approaches.

Others

Other metaheuristics applied to OMM were found by this study, however as the rate of occurrence was low, the following metaheuristics were grouped into a single group: Prey-Predator, Harmonic Search and Shuffled Frog Leaping Algorithm.

Prey-Predator algorithm (PPA) (LAUMANN; RUDOLPH; SCHWEFEL, 1998) is a metaheuristic algorithm inspired by interaction between prey and predator among animals. The first version of the PPA did not foresee the natural movement of prey, this movement occurred only if there was pressure from a predator. (TILAHUN; ONG, 2015) proposes a newer version with adaptations to work with continuous solution space. In this recent approach, the prey moves toward the best prey at each iteration, resembling the PSO. (FERRANTI; SOARES; SOUZA, 2018a) presents a study using PPA to find the best weights to aggregate a set of similarity measures. Among the nature-inspired population approaches, PPA has characteristics that are adequate for the meta-matching problem: it allows that promising regions of the solution space be exploited by a set of agents (solutions), pressured to run away from regions that are not attractive in terms of the value of the objective function, while allowing the exploration of new regions by assigning to these agents a pseudo-random behavior in the definition of their

displacement. The PPA used by (FERRANTI; SOARES; SOUZA, 2018a) is recommended for problems with continuous representation.

Harmony Search Algorithm (HSA) (GEEM; KIM; LOGANATHAN, 2001) is a music-based meta-heuristic optimization algorithm which mimics the improvisation process of jazz musicians when producing a harmony. (FORSATI; SHAMSFARD, 2016) presents HSOMap, an approach for OMM that uses a HSA in the mapping extraction step. The approach computes a similarity matrix for each similarity measure used, the mapping extraction step selects the best matches by analyzing the matrices using HSA as a combination optimizer algorithm and a correspondences-based representation. HSA has a good trade-off between exploration and exploitation that makes it appropriate for optimization problems with complex solution spaces (e.g., combinatorial) such as ontology mapping, according to the authors.

Shuffled Frog Leaping Algorithm (SFLA) (EUSUFF; LANSEY; PASHA, 2006) explores the collective behavior of solutions. The movement of solutions in the search space occurs towards the best solutions found so far. SFLA is an algorithm that resembles PSO. In SFLA, candidate solutions are divided into groups, where the worst individual is likely to move in two directions: toward the best frogs in the group or toward the best overall frog. If no move generates an enhanced solution, a new random solution is created.

Each meta-heuristic can work with different ways of representing a solution, in the next subsection the main ways of representation and their particularities are presented.

2.5.6 RQ6: How have solutions been represented?

There are many different characteristics that can influence a user or researcher in choosing a meta-matcher tool. The application scenario is a determining factor for choosing a meta-matcher. For instance, for the web query answer problem a quick response is more important than a highly accurate answer (SHVAIKO; EUZENAT, 2013). The way the problem is modeled can influence the appropriateness of a meta-matcher in specific scenarios. Modeling affects how a metaheuristic will work, which operators can be executed, and other characteristics of the approach. As Table 10 shows, most common articles use a problem modeling that represents thresholds, correspondences or weights. Meta-matching can be addressed using discrete or continuous modeling. Discrete modeling is often used to find the best set of correspondences between ontologies. Continuous modeling is often used to find the best weights to combine matchers. Both approaches can add threshold values in their representation model which is also continuous values.

Ontology meta-matching approaches focus on how to determine the optimal parameters to aggregate various similarity measures to obtain the high-quality ontology alignment (XUE; PAN, 2018). A weight-based modeling aims to find the best values to aggregate a set of similarity measures (matchers).

Table 10 – Representation models

	Threshold	No Threshold	Total
Continuous	23	8	32
Discrete	6	14	19

Source: created by the author.

There are different types of weight-based modeling, each one using different encoding mechanisms. Depending on the encoding, some evolution operators may work better than others. The simplest representation stores in a features vector each weight as a real number. Each position of the vector represents one similarity measure. (XUE; PAN, 2018) uses a variation of this representation where weights are normalized by the sum of all weights in the decoding process. Let p be the number of weights, the individual is represented by a set of numbers $\{c_1, c_2, \dots, c_p\}$ whose decoding is carried out by dividing the numbers with their sum, generating the parameter set $\left\{ \frac{c_1}{\sum_{i=1}^p c_i}, \frac{c_2}{\sum_{i=1}^p c_i}, \dots, \frac{c_p}{\sum_{i=1}^p c_i} \right\}$.

In some studies, the authors propose to manipulate only cutoff points in the feature vector (XUE; WANG; HAO, 2014; XUE; WANG; HAO, 2015). This encoding mechanism stores several cuts or separation points within the range $[0,1]$. Each cutoff point delimits the weight size. Let p be the number of weights, the set of cuts can be represented as $c' = c'_1, c'_2, \dots, c'_{p-1}$, where the c' set represents the characteristics of the individual. The individual decoding is carried out by queuing the elements of c in ascending order. The result is $c = c_1, c_2, \dots, c_{p-1}$ and the weights are calculated as follows:

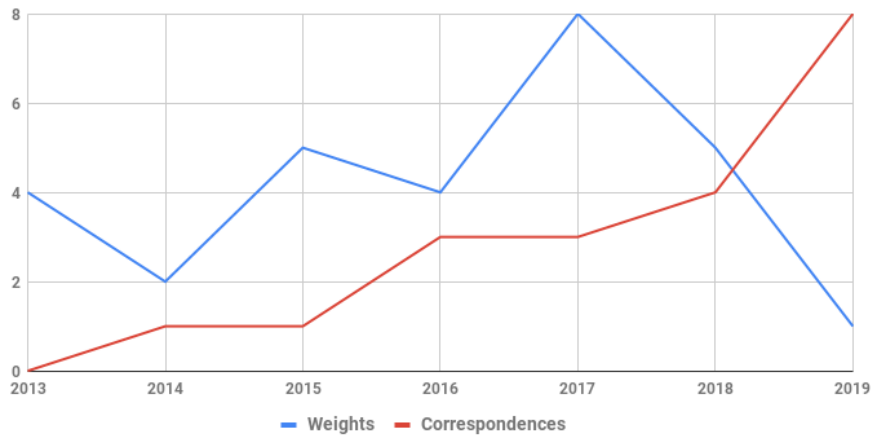
$$w_i = \begin{cases} c_1, & i = 1; \\ c_i - c_{i-1}, & 1 < i < p; \\ 1 - c_{p-1}, & i = p. \end{cases} \quad (2.1)$$

In addition to representing weights, continuous modeling can also be used to represent the similarity threshold. In our search, 32 articles have used continuous modeling, eight of them have modelled the problem using a weight-based representation with threshold, 21 articles have added the threshold in their representation, and two articles have used a representation containing threshold values only.

Whereas in weight representation the problem is modeled as function optimization, in correspondence representation the problem is modeled as a combinatorial optimization problem. In this representation, all possible match pairs are mapped to compose the individual's characteristics. Authors usually choose to map correspondences between ontology classes (MOHAMMADI; HOFMAN; TAN, 2019) or ontology instances (XUE; WANG, 2015a). Unlike weight-based modeling whose output must be processed by a meta-matcher, in the correspondences approach the output is the final set of aligned entities.

The most commonly encoding used in correspondences modeling is through discrete numbers. In this encoding, the individual is represented by a set of discrete numbers representing

Figure 9 – Number of articles using weight-based and correspondence-based modeling



Source: created by the author.

indexes for entities of the target ontology. The size of the discrete number set is defined by the cardinality of the source ontology. A specific index maps a possible correspondence between two entities, one entity from each ontology (XUE; REN, 2017; MOHAMMADI; HOFMAN; TAN, 2019). Let $|O_1|$ and $|O_2|$ be the cardinalities of two ontologies O_1 and O_2 , an individual X is defined as $X = (x_1, x_2, \dots, x_{|O_1|})$, $x_i \in \{1, 2, \dots, |O_2|\}$, $i = 1, 2, \dots, |O_1|$ where $x_i, i = 1, 2, \dots, |O_1|$, represents the i -th correspondence. $x_i = j$ means that the i -th entity of O_1 is mapped to the j -th entity of O_2 .

According to Table 10, the most popular representation is the weight-based modeling using a threshold value. The threshold value is usually independent of the weights, and it acts as a cutoff point to filter out candidate matches that have not reached the minimum confidence. The threshold ends up controlling the precision and recall scores. Figure 9 presents the data grouped by removing the threshold and maintaining the weight and correspondence models, and it points out to the growth of using correspondence-based modeling. Despite surpassing the number of weight-based approaches in 2019, 75% of the correspondence-based approaches in 2019 shares the same first author and making it difficult to determine if this trend will last. The advantage of using the weight-based representation is that once calculated, the weights can be stored and the experiment can be resumed at a specific point, i.e. the weights can be used later without the need to find them again. In short, the choice of modeling is strongly related to the application scenario and what the alignment tool aims to deliver to the user.

2.5.7 RQ7: What are the main objective functions?

The objective function (OF) is the main indicator of the quality of a solution within the process of a metaheuristic. The OMM area can be divided into systems that work with supervised or unsupervised model training. Of the 51 articles, 18 presents supervised approaches and 32

presents unsupervised approaches (both approaches are analyzed in (FORSATI; SHAMSFARD, 2016)).

In supervised systems, where part of the answer (correspondences) is embedded in the system input, it is common to compare the result found by a solution with the reference (small correspondence set) result. In these systems, traditional information retrieval measures are used: precision (rate of correct correspondences returned), recall (the rate of expected correct correspondences returned) and f-measure (harmonic mean between precision and recall). These metrics were used as OF in 89% of the supervised systems, such as in (XUE; WANG; HAO, 2014; MARJIT, 2015; BINIZ; AYACHI, 2018) and others. In (FERRANTI; SOARES; SOUZA, 2018a), the reference correspondences are used in the construction of a linear system whose objective function seeks to find the best solution to solve the system.

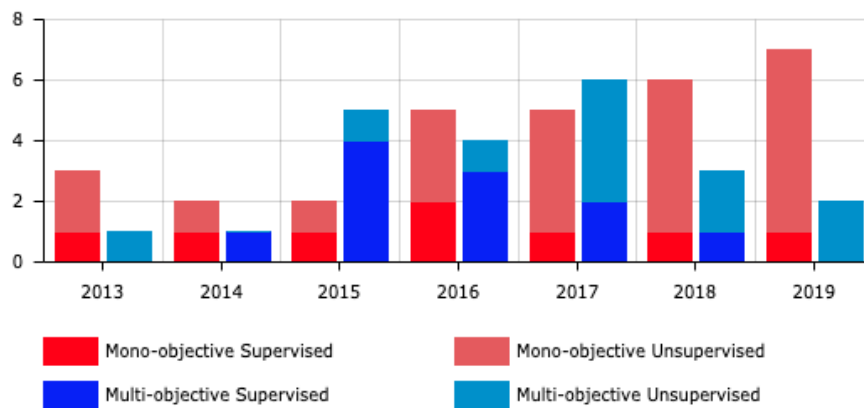
On the other hand, there are more metrics used as OF in unsupervised approaches due to the lack of reference for system training, causing the authors to develop and experiment different ways to guide the algorithm convergence. Several authors have used OFs that can replace precision, recall and f-measure, and some of the metrics found are MatchRatio, MatchCoverage and MatchFmeasure.

MatchCoverage is a substitute for recall, it represents the fraction of entities which exist in at least one correspondence in the resulting alignment in comparison to the total number of entities in the ontology (XUE; WANG, 2015b). MatchRatio is a substitute for precision, is the relationship between the number of correspondences found and the number of the corresponding entities. The premise is that the precision of a match result is better if an entity is not vaguely corresponds to many other concepts but only less (XUE; WANG, 2015b). Similarly, MatchFmeasure is the harmonic mean between MatchCoverage and MatchRatio. About 52% of the unsupervised systems use MatchFmeasure, MatchCoverage or MatchRatio.

Unlike precision, recall and f-measure based OFs, other researchers explore the confidence value of matchers and the cardinality of resulted alignment. In the first case, the objective function evaluates the average correspondences similarity, the higher the average correspondences similarity in a solution, the more likely the candidate alignment has to be correct. In the second case, the objective function evaluates the number of correspondences in a candidate solution, calculating the distance between the number of the solutions and the quantity defined by the researcher. Usually, the quantity is equals to the cardinality of the smallest ontology. OFs based on alignment cardinality and mean similarity are generally used together.

The objective functions used in the 51 articles can also be classified into mono or multi-objective approaches. Figure 10 presents the evolution of approaches both in terms of supervision and the type of objective function. Figure 10 shows a increasing interest in multi-objective approaches between 2015 and 2017. Shortly after the draw between supervised and unsupervised in 2016, the growing number of unsupervised systems indicates that researchers are developing increasingly robust and independent systems.

Figure 10 – Mono-objective/multi-objective and supervised/unsupervised approaches over the years.



Source: created by the author.

2.5.8 RQ8: Which datasets has been used for OMM evaluation?

The Ontology Alignment Evaluation Initiative⁷ (OAEI) is an international initiative to coordinate the evaluation of computational approaches that address the ontology matching problem. Several datasets can be found on the OAEI website, each one with ontologies that represent the main challenges in the OM area, such as complex alignments, large-scale ontology matching, and others. OAEI datasets are often used for OMM evaluation as well. The complete list of OAEI datasets used in the 51 articles is presented in Table 11.

Table 11 – OAEI Datasets

Dataset	Total
Benchmark (B)	31
Anatomy (A)	14
Large Biomedic (LB)	12
Library (L)	5
Conference (C)	5
Disease and Phenotype	2
HOBBIT Link Discovery	1
Instance matching	1

Source: created by the author.

The benchmark test is the most popular dataset for OMM evaluation. The OMM area does not have a specific test dataset; however, the benchmark's characteristics are adequate to evaluate OMM approaches. According to the benchmark's webpage⁸, the goal of the dataset is to offer a set of tests which are wide in feature coverage, progressive and stable. The benchmark contains several test cases which are systematically generated to start from some reference

⁷ <http://oaei.ontologymatching.org/>

⁸ <http://oaei.ontologymatching.org/2016/benchmarks/index.html>

ontology and discarding a number of information in order to evaluate how an ontology matcher behave when this information is lacking. This benchmark is best suited for OMM because as the tests alter or remove information from the target ontology, the algorithm is required to self-adjust so that other similarity measures are used according to its performance in that scenario. However, some authors have also evaluated their OMM approaches on other datasets. In this section, the articles are discussed into two groups: (i) approaches evaluated using real ontologies and (ii) approaches using synthetic ontologies.

Among approaches in the first group, stands out the use of sensor ontologies such as the Semantic Sensor Network Ontology⁹ (SSN). Some authors (XUE; PAN, 2018; XUE; CHEN, 2018; XUE; CHEN, 2019c) have demonstrated the effectiveness of their approaches using the SSN and other sensor ontologies, such as the CSIRO¹⁰ and the MMI¹¹. Unlike the others, (XUE; LIU, 2017b) seeks to align the FreeBase¹², YAGO¹³ and IMDb¹⁴ ontologies.

Other authors (KUREYCHIK; SEMENOVA, 2017; SEMENOVA; KUREYCHIK, 2016b; MARJIT, 2015; KESSENTINI et al., 2014; RAMESH et al., 2016; MARTÍNEZ-ROMERO et al., 2013; SEMENOVA; KUREYCHIK, 2016a) have created their own ontologies. However, it is difficult to compare these results with other approaches, because the ontologies are generally not available for further research.

The diversity of datasets used shows that the OMM area lacks a robust and adequate dataset to support OMM research. OAEI datasets are still the main reference for comparing OMM approaches. On the other hand, many authors do not submit their code to OAEI annual campaigns, nor do they disclose the detailed results of the approaches. Table 12 contains the f-measure values of each of the papers that used one or more OAEI datasets. The Large Bio (LB) dataset has been subdivided into three categories: LB1 (FMA¹⁵ and NCI¹⁶ mappings), LB2 (FMA and SNOMED¹⁷ mappings), and LB3 (SNOMED and NCI mappings). Each category represents tasks with specific ontologies and a specific f-measure. It is possible to observe the predominance of the Biblio benchmark. Several approaches were able to reach f-measure rates around 90%, which may be an indication of search saturation in this dataset that was not created exclusively for evaluating meta-matchers.

⁹ <https://www.w3.org/TR/vocab-ssn/>

¹⁰ <https://www.w3.org/2005/Incubator/ssn/wiki/SensorOntology2009>

¹¹ <https://mmisw.org/ont/mmi/device>

¹² www.freebase.com

¹³ <https://github.com/yago-naga/yago3>

¹⁴ <https://www.imdb.com/interfaces/>

¹⁵ <http://sig.biostr.washington.edu/projects/fm/>

¹⁶ <http://ncit.nci.nih.gov/>

¹⁷ <http://www.ihtsdo.org/index.php?id=545>

Table 12 – F-measure values in OAEI datasets.

	B	A	L	LB1	LB2	LB3	C	Others
(XUE; PAN, 2018)	0.90	0.95	0.84	0.89	0.90	0.78	-	-
(XUE et al., 2015)	0.94	-	-	-	-	-	-	-
(XUE; WANG; HAO, 2014)	0.93	-	-	-	-	-	-	-
(XUE; CHEN, 2018)	0.92	0.95	-	-	-	-	-	-
(ACAMPORA et al., 2013a)	0.90	-	-	-	-	-	-	-
(XUE; WANG; HAO, 2015)	0.94	-	-	-	-	-	-	-
(XUE; WANG, 2015a)	0.92	0.92	0.83	-	-	-	-	-
(XUE; LIU, 2017a)	-	0.71	-	0.68	-	0.79	0.41	-
(XUE; TSAI; ZHANG, 2016)	0.94	-	-	-	-	-	-	-
(JIANG; XUE, 2015)	0.96	-	-	-	-	-	-	-
(XUE et al., 2015)	0.90	-	-	-	-	-	-	-
(XUE; WANG, 2017)	0.94	-	-	-	-	-	-	-
(XUE; LIU, 2017c)	0.93	-	-	-	-	-	-	-
(FERRANTI; SOARES; SOUZA, 2018a)	0.90	-	-	-	-	-	-	-
(XUE; TSAI; WANG, 2017)	0.94	-	-	-	-	-	-	-
(XUE, 2017)	0.97	-	-	-	-	-	-	-
(XUE; REN, 2017)	0.73	-	-	-	-	-	-	-
(XUE et al., 2018a)	-	0.95	-	0.88	0.85	0.80	-	-
(XUE; TSAI; FENG, 2017)	0.91	-	-	-	-	-	-	-
(XUE; CHEN, 2019c)	0.65	0.96	0.88	0.91	0.90	0.78	-	-
(XUE; HANG; TANG, 2019)	-	0.97	-	0.97	-	0.86	-	-
(XUE; LU; CHEN, 2019)	-	0.96	-	0.90	0.87	0.82	-	-
(XUE; CHEN; YAO, 2018)	-	0.96	-	0.96	-	0.85	0.84	-
(MOHAMMADI; HOFMAN; TAN, 2019)	-	0.87	-	-	-	-	0.72	0.75 (Disease and Phenotype DOID-ORDO) 0.73 (Disease and Phenotype HP-MP)
(XUE; WANG, 2015b)	0.79	-	-	-	-	-	-	-
(FORSATI; SHAMSFARD, 2016)	0.76	-	-	-	-	-	-	-
(XUE; TANG, 2017)	0.88	-	-	-	-	-	-	-
(XUE; WANG; REN, 2014)	0.86	-	-	-	-	-	-	-
(XUE et al., 2018b)	-	0.95	-	0.88	0.82	0.79	-	-
(ACAMPORA et al., 2013b)	0.86	-	-	-	-	-	0.43	-
(JIANG et al., 2016)	0.96	-	-	-	-	-	-	-
(XUE; CHEN, 2019a)	-	-	-	-	-	-	-	0.85 (Disease and Phenotype DOID-ORDO) 0.89 (Disease and Phenotype HP-MP)
(XUE; CHEN, 2019b)	0.61	0.94	0.85	0.89	0.90	0.78	-	-
(XUE; LIU, 2018)	-	-	-	-	-	-	-	1.00 (HOBBIT Link Discovery)
(XUE; CHEN; CHEN, 2019)	-	0.95	-	0.88	0.86	0.82	-	-
(XUE; PAN, 2017)	0.81	0.89	0.82	0.85	0.78	0.72	-	-
(XUE; LIU, 2017b)	-	-	-	-	-	-	-	0.98 (Instance Matching author-dis) 1.00 (Instance Matching author-rec) 0.97 (Instance Matching val-sem) 0.97 (Instance Matching val-struct) 0.96 (Instance Matching val-struct-sem)
(GULIĆ; VRDOLJAK; PTIČEK, 2018)	0.90	-	-	-	-	-	-	-
(XUE; LIU; WANG, 2017)	0.89	-	-	-	-	-	-	-
(ZHANG; GUO, 2019)	0.77	-	-	-	-	-	0.67	-
(XUE; WANG; REN, 2016)	0.81	-	-	-	-	-	-	-
(ACAMPORA; LOIA; VITIELLO, 2013)	0.94	-	-	-	-	-	-	-
(XUE et al., 2018c)	-	-	-	0.89	0.84	0.79	-	-

Source: created by the author.

2.6 THREATS TO VALIDITY

This systematic literature mapping aimed to identify challenges in ontology meta-matching domain. However, as any method, there are threats to its validity and limitations. The results of this study may have been influenced by certain uncontrollable limitations.

Although the mapping covered 51 articles, removing articles not written in English and those in gray literature, for example, diminished the accuracy of the conclusions. During the selection process some influential work in the area may have been lost, however this mapping focused on articles whose explanation of the main components of the OMM process was clearly discussed. Furthermore, we did not consider all the relevant electronic databases, i.e. ACM Digital Library, so it is possible that relevant studies were not indexed in the selection of this mapping. However, this research relies on the representativeness of the repositories selected to answer the research questions.

The systematic research criteria used in this paper may also be subject to criticism, threatening the validity of the final result. The search string may not contain all relevant keywords, causing the loss of some studies and errors may be entered in the protocol definition. To minimize this, the search string was evaluated using articles to track results. The articles appeared in the results generating evidence about the correctness of the research.

Finally, all data and spreadsheets generated in this work are available for further research¹⁸.

2.7 CONCLUDING REMARKS

This paper presents a systematic mapping in ontology meta-matching. The mapping was performed through a reproducible procedure using a search string that was created from PICO terms. Initially 1137 articles from three different online repositories were found. After filtering articles using inclusion and exclusion criteria, the final amount of 51 articles was used to answer a set of research questions pertinent to the ontology meta-matching (OMM) area.

The ontology meta-matching area has evolved in the recent years. Evidence of this evolution is the growth of approaches that deal with more complex versions of the problem, such as unsupervised approaches (Figure 10) and meta-matching approaches applied in large scale ontologies (Table 12). Despite advances in this area, many authors do not submit their systems for the annual OAEI campaigns, which is the main reference to evaluate the progress of research in the area. The lack of a specific dataset to evaluate meta-matching approaches contributes to the low number of OMM systems that are submitted to the annual campaigns. As seen in this mapping, OMM researchers evaluate their systems using OAEI datasets but do not participate in the campaigns. It would be interesting to create a specific track for OMM approaches to better

¹⁸ <https://github.com/lapic-ufjf/OMM-systematic-mapping>

compare other aspects of these systems beyond accuracy.

There are still issues that need to be resolved for systems to become more popular and to be effectively applied in real world scenarios. For alignment systems to gain broad acceptance and be reliable for users, they need to provide explanations of their results to users or other programs that exploit them. Reliability and provenance issues become particularly important in web configurations that enable social and collaborative matching (SHVAIKO; EUZENAT, 2013). Few papers are concerned with explaining results; results usually contain only precision and recall, leaving aside important issues to point out the relevance of the research. The meta-matching process can be subdivided and has several execution steps until a solution is built, comparative analysis between similarity measures is an example of relevant information that is little discussed in the articles. In general, articles that work with domain-specific ontologies tend to explore the similarity measure factor because it is used a domain-specific measure. This systematic mapping has verified that there is a set of measures frequently used by most authors, however no performance analysis of these measures was conducted.

This section has mapped out research questions that seek to cover the main aspects of ontology meta-matching and each aspect has its specificities. OMM is a multidisciplinary problem and some aspects related to the area of Computational Intelligence may be analyzed in greater detail in the future, such as the behavior of metaheuristics, the performance of objective functions, and how each approach explores the search space. This article did not discuss the runtime factor of each system.

3 MANUSCRIPT 2: A FRAMEWORK FOR OMM EVALUATION

Ontology matching has become a key issue to solve problems of semantic heterogeneity. Several researchers propose diverse techniques that can be used in distinct scenarios. Ontology meta-matching approaches are a specialization of ontology matching and have achieved good results in pairs of ontologies with different types of heterogeneities. However, developing a new ontology meta-matcher can be a costly process and a lot of experiments are often carried out to analyze the behavior of the matcher. In this master's thesis, the main objective is to evaluate costs and contributions of algorithms commonly applied to the OMM problem. This chapter is a full article that presents the implementation details of a framework designed to simplify the process of developing and testing OMM approaches, an important contribution to this research since it makes it possible to test different algorithms at different stages of the workflow. This framework aims to aid researchers to develop and analyze algorithms for ontology meta-matching, mainly metaheuristic-based supervised and unsupervised approaches. As the main contribution of the chapter, the framework proposed will facilitate the evaluation of ontology meta-matching approaches and, as the secondary contribution, a data provenance model that captures the main information generated and consumed throughout experiments is presented in the framework. Section 3.2 of this chapter contains text presented earlier in Section 1.2.

3.1 INTRODUCTION

Throughout the evolution of Web, several challenges have emerged to optimize information exchange and facilitate the consumption of Web-published data. Ontologies have emerged to determine a well-defined meaning for data and to reduce semantic heterogeneity problems when this data is consumed (SOUZA; SIQUEIRA; NUNES, 2019). However, ontologies themselves have become the cause of a new problem in data semantics: using more than one ontology in an application can lead to ambiguity in data interpretation, mainly because ontologies are built by engineers with different needs and domain views (ACAMPORA et al., 2013a).

The ontology matching (OM) problem consists of finding relationships between entities of different ontologies (FARIA et al., 2013). Once multiple relationships are established, knowledge of ontologies can be used together without semantic heterogeneity, facilitating the development of increasingly intelligent applications.

Although the OM problem is relatively recent, several approaches have already been proposed to find relationships through the use of similarity measures. There are a variety of similarity measures applicable to this problem, such as JaroWinkler (WINKLER, 1999), WordNet-based distance (BUDANITSKY; HIRST, 2001), or the similarity flooding algorithm (MELNIK; GARCIA-MOLINA; RAHM, 2002). Each of these measures is usually used to compute the similarity between ontology entities under different aspects. In (EUZENAT; SHVAIKO, 2013), the authors present a detailed taxonomy of similarity measures. As there is no better

similarity measure than the others for all cases (XUE; TANG, 2017), a combination of measures is necessary to create general-purpose ontology matchers. Ontology meta-matching (OMM) is a subarea of ontology matching that aims to address the OM problem by combining different similarity measures. OMM approaches use methods to select the correct matching components to execute, and to adjust the multiple knobs (e.g. threshold, coefficients, weights, etc.) of components that are part of the OMM process (XUE; PAN, 2018). Several algorithms can be used in the meta-matching task however, modeling the problem as an optimization problem and using meta-heuristics to solve it is becoming the state of art in the area (XUE; CHEN, 2018). Heterogeneous ontology integration problems belong to a class of NP-hard optimization problems, and can be solved by evolutionary algorithms (KUREYCHIK; SEMENOVA, 2017).

Due to the size of the OMM task and the wide variety of solutions that can be employed at each processing step, developing a completely new approach can be a costly task. A researcher must be familiar with APIs and their structures for representing ontologies, with the state of the art in similarity measures as well as the cost/benefit ratio of each. Moreover, it is necessary to develop optimization and selection algorithms that are analyzed through a battery of experiments. The goal of this paper is to propose a modularized OMM framework that facilitates the process of developing and carrying out experiments with distinct approaches. The framework was designed to metaheuristic-based ontology matchers and allows to develop supervised or unsupervised approaches. In order to exemplify the use of the framework, some of the main algorithms used in each step of the matching process have been implemented and are available to researchers. The main contribution of this framework is to facilitate the development and evaluation of OMM approaches, covering the main stages of this kind of experiment. The secondary contribution is the inclusion of a data provenance model that captures the main information generated and consumed throughout the experiment.

The remaining of this paper is organized as follows. Section 3.2 formalizes the meta-matching problem and presents the main definitions. Section 3.3 presents the related works. Section 3.4 presents our approach proposed and explains how researchers can couple their algorithms and develop their own solutions. Section 3.5 details the experiments demonstrating the usability of the framework to verify meta-heuristic and objective function behaviors and, finally, Section 3.6 discusses the results and lessons learned through this study and points to future works.

3.2 PROBLEM STATEMENT

An ontology is a formal and explicit specification of a shared conceptualization (GRUBER, 1993a), where "formal" means machine readable, "explicit specification" refers to concepts, properties, relations, functions, restrictions, axioms, defined explicitly; "shared" means consensual knowledge; and conceptualization refers to an abstract model of some phenomenon of the reality. The matching task between two ontologies (source and target) consists of the search

for semantic relationships (correspondences) between entities of source ontology with entities of target ontology (ACAMPORA et al., 2013a). The term entity in this case is broader than it is in classic database models, i.e., the entity-relationship model; in an ontology all objects are first-class citizens, this includes classes, relationships, instances, data types, and values.

Definition 5. (Similarity measure). *Let E_1 be the set of entities from o_1 and E_2 be the set of entities from o_2 , where $o_1, o_2 \in O$ and O is a set of ontologies. A similarity measure $sm : E_1 \times E_2 \rightarrow \mathbb{R}$ assesses the similarity between two entities $e_1 \in E_1$ and $e_2 \in E_2$ into a score $\eta_{1,2}$ where $\{\eta_{1,2} \in \mathbb{R} | 0 \leq \eta_{1,2} \leq 1\}$. An entity similarity measure is also called matcher.*

There are different types of relationships. An entity $e_i \in E_1$ can be equivalent, disjoint, or subsumes an entity $e_j \in E_2$, for example. In the context of this article, these relationships are limited to equivalence relationships.

Definition 6. (Alignment). *The alignment is the output of the matching process. An ontology alignment A is a set of tuples (or correspondences) in the form $(e_i, e_j, \diamond_{i,j}, \eta_{i,j})$ where e_i and e_j represents two entities from distinct ontologies, followed by the relation descriptor \diamond representing a relationship type and $\eta_{i,j}$ is the confidence value of the correspondence.*

Definition 7. (Ontology matching task). *Let O be a set of ontologies and A a set of alignments. An ontology matching task $omt : O \times O \rightarrow A$ maps $o_1 \in O$ and $o_2 \in O$ to an alignment $a \in A$ through the use of a similarity measure.*

The term meta-matching systems was presented by (EUZENAT; SHVAIKO et al., 2007) to describe systems that automatically parametrize a set of similarity measures (matchers). A meta-matcher approach deals with a set of candidate correspondences provided by several single matchers and is intended to combine the results and extract the most probable alignment.

Definition 8. (Ontology Meta-Matching task). *Let SM be the set of similarity measures. A ontology meta-matcher $omm : SM \rightarrow A$ combines the similarity score of each $sm \in SM$ for each pair $\langle e_i, e_j \rangle$ where $e_i \in E_1$ and $e_j \in E_2$ into a unified score $\eta_{e_i, e_j} \in \mathbb{R}$ where $\{\eta_{e_i, e_j} \in \mathbb{R} | 0 \leq \eta_{e_i, e_j} \leq 1\}$ produces the best alignment $a \in A$.*

Finding the best set of parameters to combine the similarity measures is a complex task and it is currently addressed through the use of meta-heuristics (XUE; CHEN, 2019b; SEMENOVA; KUREYCHIK, 2016b; SOUZA; SIQUEIRA; NUNES, 2019). Each meta-heuristic uses one or more objective functions to evaluate the quality of a solution. The solutions are proposed using a supervised (using a reference alignment) or an unsupervised approach. The main goal is to define the final confidence value for each candidate match through intelligent exploration of the search space. The final set of candidate correspondences is then subjected to a process to select the matches that have obtained the highest confidence rates. The set of selected correspondences is the answer to the problem; this set is called alignment. Some solutions,

however, are modeled in such a way that the correspondence selection process is unnecessary (XUE et al., 2018a; MOHAMMADI; HOFMAN; TAN, 2019).

To evaluate the result of the OMM algorithm, the most commonly used measures are Precision and Recall metrics (MANNING; RAGHAVAN; SCHÜTZE, 2008). These metrics compare the obtained alignment with the expected alignment. In some application scenarios such as the query answering, measures that assess processing time become more relevant to the experiment (SHVAIKO; EUZENAT, 2013).

The development of OMM approaches, as any other machine learning approach, requires a battery of experiments in order to refine the solution. Throughout these experiments, various informations regarding the execution of the process is generated and much of this information is lost at the end of the experiment. Much is discussed about how to make the results more reliable and how to make the experiment consolidated. According to (SHVAIKO; EUZENAT, 2013), for matching systems to gain broader acceptance and be trusted by users, they will need to provide explanations of their results to users or other programs that exploit them. Reliability and provenance issues become particularly important in Web configurations that allow social and collaborative correspondence. Data provenance is the record of the history of data derivation, which enables the interpretation of results, as well as reproducibility and diagnosis of problems (LIM et al., 2010). The problem of systematically capturing and managing the provenance of computational tasks is relevant to a wide range of domains and applications (FREIRE et al., 2008). In scientific experiments, provenance helps to interpret and understand results: by examining the sequence of steps that led to a result, the researcher can better understand the chain of reasoning used in its production, verify that the experiment was carried out according to acceptable procedures, identify the experiment inputs and, in some cases, reproduce the result (FREIRE et al., 2008). The modeling proposed by this article contemplates the main stages of the meta-matching process and captures the main data generated in each stage through a provenance model.

3.3 RELATED WORK

Ontology matching research is relatively recent, but has a large scope of development. In (OTERO-CERDEIRA; RODRÍGUEZ-MARTÍNEZ; GÓMEZ-RODRÍGUEZ, 2015), the authors highlight the growth in the number of articles published in the last decade. New subareas have emerged due to the need to solving increasingly complex problems, such as large-scale ontology matching and ontology meta-matching (OMM). Large-scale matching involves ontologies with tens of thousands of classes, which raises the challenge of working with large volumes of data. The solutions usually use a partitioning approach to identify similar subgraphs or clusters in the ontology before performing the matching (OCHIENG; KYANDA, 2018; XUE; PAN, 2017). The complex matching (THIÉBLIN et al., 2019) is considered as one of the main challenges of OM. The focus is to identify more complex relationships capable of representing other types

of semantic relationships besides the 1:1 equivalence. Complex matching approaches often set atomic patterns used to detect logical relations as well as transformation functions (ROUCES; MELO; HOSE, 2016; CHONDROGIANNIS et al., 2014).

The OMM problem can be addressed in several ways, however, modeling the problem as an optimization problem and using meta-heuristics to solve it is becoming the state of the art (XUE; CHEN, 2018). Among the most common meta-heuristics used are the Genetic Algorithm (GA) (XUE; PAN, 2018; XUE; LU; CHEN, 2019), the Memetic Algorithm (ACAMPORA; LOIA; VITIELLO, 2013; RAMESH et al., 2016) and the Particle Swarm Optimization (PSO) (SEMENOVA; KUREYCHIK, 2016b; MARJIT, 2015). Some authors argue that non-population metaheuristics can reduce the costs of carrying out the experiment (MOHAMMADI; HOFMAN; TAN, 2019). On the other hand, some authors present variations of populational algorithms with optimized performance (XUE; LIU, 2018).

Another aspect with influence in the construction of a solution is the objective function used in the meta-heuristic. The objective function (OF) is the main indicator of the quality of a solution within the process of a metaheuristic. The OMM area can be divided into systems that work with unsupervised or supervised model training. In (ACAMPORA et al., 2013a; XUE; CHEN, 2019c; XUE; LU; CHEN, 2019), the authors use the maximization of the sum of the confidences attributed to each correspondence as objective function, that is, the more correspondences in an alignment, the better the sum of the confidences. On the other hand, to reduce the number of incorrect matches, some authors insert a minimum confidence threshold and include an opposite objective function, leading to the creation of some multi-objective approaches to the OMM problem. Recently, MatchFmeasure have been used as objective function (XUE; PAN, 2018; XUE; LIU, 2018). MatchFmeasure estimates the quality of the alignment by analyzing the number of entities present in the alignment, in relation to the number of entities on each ontology and in relation to the amount of correspondences present in the alignment. Both confidences maximization and MatchFmeasure are unsupervised approaches, which do not require reference alignment for execution. The literature also presents OMM tools using supervised approaches as objective functions. In supervised systems, where part of the answer (correspondences) is embedded in the system input, it is common to compare the result found by a solution with the reference ontology. In these systems, traditional information retrieval measures are used: precision (rate of correct correspondences returned), recall (the rate of expected correct correspondences returned) and f-measure (harmonic mean between precision and recall), as in (XUE; WANG; HAO, 2014; MARJIT, 2015; BINIZ; AYACHI, 2018). In (SOUZA; SIQUEIRA; NUNES, 2019), the reference ontology is used in the construction of a linear system whose objective function seeks to find the best solution to solve the system. Due to the different types of meta-heuristics and objective functions, frameworks for comparing different proposals are important to understand what actually happens in each approach.

The use of frameworks to perform ontology meta-matching is not new in the area.

Approaches like the MaF (MARTINEZ-GIL; NAVAS-DELGADO; ALDANA-MONTES, 2012) and the AML (FARIA et al., 2013) emerged in times when the volume of articles for this problem grew year after year, as pointed out by (OTERO-CERDEIRA; RODRÍGUEZ-MARTÍNEZ; GÓMEZ-RODRÍGUEZ, 2015).

Looking for a friendly user interface, the MaF (MARTINEZ-GIL; NAVAS-DELGADO; ALDANA-MONTES, 2012) comes with a framework that can be used by both developers and front-end users. Although simple, the front-end interface for users brings the solution closer to the real world applications, partly attending to some discussions on how to popularize and increase confidence in the OMM approaches (SHVAIKO; EUZENAT, 2013). Several matchers can be used in the MaF, including hybrid matchers where atomic matchers are combined (hybrid matchers are also explored in other works (SOUZA; SIQUEIRA; NUNES, 2019)). The variety of combinations of algorithms implemented in the tool makes scientific experimentation through the MaF customizable and easily performed from the point of view of the front-end user. However, some components that are part of the system kernel can make it difficult to use by developers who want to implement more complex algorithms for a specific task. The MaF, like many other works, uses the Jena API (MCBRIDE, 2002) to create the computational structures of the ontologies, but when it is necessary to work with large-scale ontology matching, some authors opt for more optimized structures (FARIA et al., 2013).

The AML (FARIA et al., 2013) is a framework for large-scale ontology matching, and it works with ontologies in the order of tens of thousands of classes. Unlike the MaF, AML does not have a front-end user interface, being a tool more geared towards researches developers. Both approaches divide the execution of the process into steps, each step composing a framework module. The modular architecture, according to the tool implementation, allows the experimentation of algorithms that perform only a specific task, reducing the effort required by the developer. Tools like MaF, AML and GNOSiS+ (SOUZA; SIQUEIRA; NUNES, 2019) have a well-defined workflow that can be modularized. GNOSiS+ is the framework that most resembles the one proposed by this work. In GNOSiS+, several matchers can be combined through a genetic algorithm (GA) modeled in a supervised approach. The GA seeks the best parameters to combine the matchers predetermined by the researcher. The matchers used in the experiment are defined through a configuration file and can be easily replaced. However, the other steps of the process, such as the method of combining the matchers, are not configurable, limiting the use of the tool to the GA implementation.

The main output of each of these tools is the correspondences set called alignment between the ontologies, and the quality of the alignment is generally evaluated through precision and recall measures. In GNOSiS+, the output also delivers the parameters found during the execution to facilitate the reproduction of the experiment. In order to promote different comparative analyses, MELT (HERTLING; PORTISCH; PAULHEIM, 2019) presents different tools to assess the similarity between the results of the matchers. MELT is a toolkit for ontology

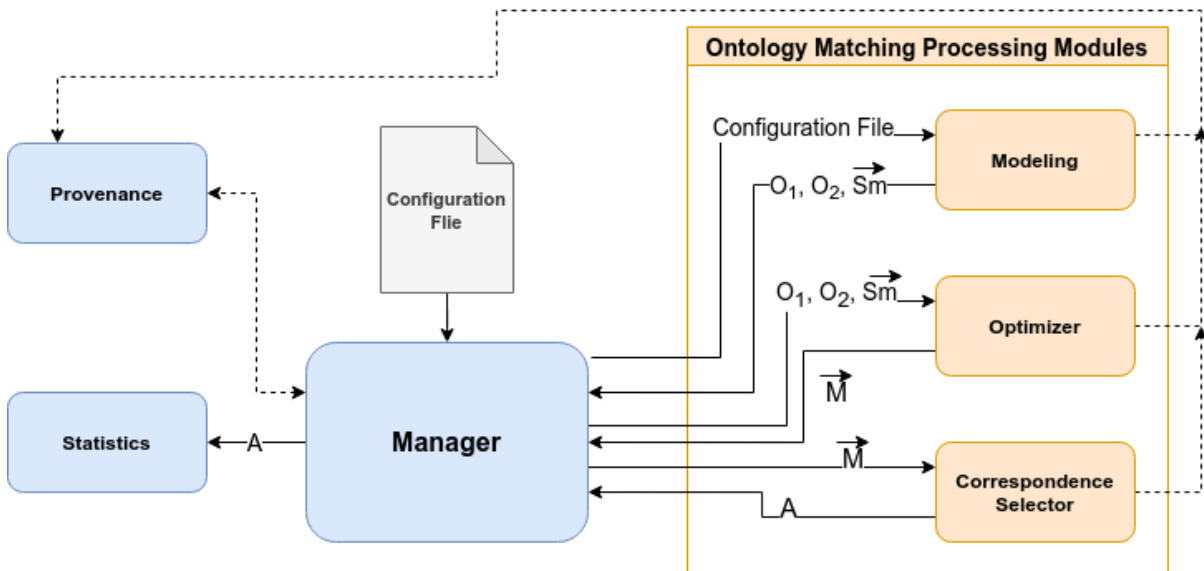
meta-matching whose components have been used individually in other researches (PAULHEIM, 2019). It allows a matcher to be coupled and evaluated by existing techniques in the tool. Its main characteristics are associated with the visualization of results and the integration with the tools provided by the OAEI, such as SEALS and HOBBIT.

Based on the solutions found in the literature, the ontology meta-matching experiment can be divided into three stages: Modeling, Optimization and Selection of correspondences. The vast majority of works presented in this section have efficient methods for the modeling stage, where the ontologies are computationally structured as well as the matchers that will be used. However, abstract solutions that encourage the development of intelligent solutions for the optimization and selection stages are little explored. This work proposes a framework that provides tools for the researcher to create intelligent algorithms in the three main stages of experimentation in ontology meta-matching, unlike MaF which provides ready-to-use algorithms. While the MaF is a solution for ontology matching that allows the user to test various matchers with predefined combination methods, our framework presents itself as an approach to meta-matching which gives the user freedom to implement different combination methods, search algorithms, and new methods for OMM. The framework proposed in this work allows the researcher to develop both supervised and unsupervised optimization algorithms, a feature not foreseen in the MaF and that in GNOSiS is restricted to supervised algorithms. Unlike MELT, where a matching approach is coupled and submitted to the available assessment measures, our framework aims to provide a common development environment so that the approaches are developed within the framework itself, making the comparison between experiments show more accurate results because the execution is within the same environment. The data provenance collector presented in this framework allows to compare experiments in a simplified way and with greater wealth of details, describing the research evolution.

3.4 PROPOSED APPROACH

In this work, we propose a framework to test ontology meta-matching approaches. It was built based on the division of ontology meta-matching process into steps, where a set of computational solutions is available for each step. The framework covers the three main steps of the OMM process: the Modeling, Optimization, and Correspondence Selection steps (see Figure 11). Each step constitutes a system module that includes algorithms to evaluate similarity between entities, optimization algorithms for combining similarity measures and selection algorithms to choose correspondences. Each module operates independently having its input data and a predefined output format. This architecture allows new approaches to be implemented at specific process steps without major difficulties. The following subsections describe each of the three processing modules including a management module, a provenance collector and a statistics generator.

Figure 11 – System architecture, representing the execution flow that starts in the Manager and ends in the analysis of the data using the Statistics module. The dashed line represents the provenance data flow while the continuous line represents the flow of the data that will be consumed for the alignment construction. Orange colored blocks represent the processing modules, responsible for carrying out the activities that will create the final alignment.



Source: created by the author.

3.4.1 Manager

An ontology meta-matching experiment consists of several processing steps and has several parameters defined by the researcher to configure the execution. This information needs to be managed and consumed at specific steps. The manager module is the first to run during the course of an experiment. The manager is responsible for controlling the flow of the experiment, managing the input and output of each module and ensuring the execution of the experiment. An experiment needs to define a set of parameters that characterize the framework's execution settings. These configurations include the pair of ontologies (O_1 and O_2) that will be aligned, a similarity measures set (Sm) that will be used to evaluate the similarity of entities, a training reference alignment set (optional parameter), the selected optimization meta-heuristic and, finally, the correspondence selection algorithm. The Manager module has as input all settings used in the execution of an experiment.

3.4.2 Modeling

The modeling module is the first to be triggered by the manager module. This module has as input the files containing the pair of ontologies that will be aligned and the name of each similarity measure that will be used in the experiment. This framework was developed to process ontologies described in OWL or RDF. All this information is read and structured in

computational objects that will be consumed in the following steps by means of the Jena API (MCBRIDE, 2002). The output consists of objects containing the ontologies data and a set of similarity measures.

The user can implement his own similarity measures as well as use the measures available in the tool. Each similarity measure (matcher) generates a matrix with $|e_1|$ rows and $|e_2|$ columns, where e_1 represents the set of entities from the source ontology and e_2 the set of entities from the target ontology. Each cell in the matrix represents the confidence value assigned by that matcher for a specific candidate match. To include new similarity measures, the user simply implements the interface that best suits the data that will be needed in the measure. For example, a String-based measure (EUZENAT; SHVAIKO, 2013) can be added using the *IEditDistance* interface because it consumes two strings, while measures such as Similarity Flooding (MELNIK; GARCIA-MOLINA; RAHM, 2002) must use the *ISimilarityFunction* interface because it is necessary to use other information from the concept. Table 13 shows the main measures available and their classification according to (EUZENAT; SHVAIKO, 2013).

Table 13 – Similarity Measures Available

Name	Classification	Reference
Wu and Palmer	Linguistic-based	(WU; PALMER, 1994)
Lesk	Linguistic-based	(BANERJEE; PEDERSEN, 2003)
Resnik	Linguistic-based	(RESNIK, 1995)
Leacock and Chodorow	Linguistic-based	(LEACOCK; CHODOROW, 1998)
Similarity Flooding	Taxonomy-based	(MELNIK; GARCIA-MOLINA; RAHM, 2002)
JaroWinkler	String-based	(WINKLER, 1999)
Levenshtein	String-based	(LEVENSHTein, 1966)
DamerauLevenshtein	String-based	(DAMERAU, 1964)

Source: created by the author.

3.4.3 Optimization

The optimization step is the experiment step responsible for combining multiple similarity measures and defining a confidence value for each pair of ontology entities. An optimizer can be defined as a tuple $\sigma = (o_1, o_2, Sm, HSA)$ where:

- o_1 and o_2 are the computational structures of the ontologies to be matched, represented by an instance of type Model that contains the methods to retrieve the pertinent information of the ontology data.
- Sm represents the similarity measures set. For each $sim \in Sm$, $sim(e_1, e_2) = Conf_{sim}$ where e_1, e_2 are entities from distinct ontologies and $Conf_{sim} \in [0, 1]$.
- HSA is an heuristic search algorithm.

To perform the optimization process, it is common in the literature to use metaheuristics such as Genetic (XUE; PAN, 2018), Memetic (XUE; CHEN, 2019b), PSO (SEMENOVA; KUREYCHIK, 2016b) algorithms and many other approaches as search algorithm. Evolutionary algorithms are becoming the state of the art in the search for solutions to this problem (XUE; LIU, 2017b). These search heuristics are usually used to find the best combination of similarity measures. Despite having inspiration in different phenomena, the sequence of operations of meta-heuristics are similar, generally making use of (1) a phase of building one or more solutions, (2) diversification of solutions to explore the search space and (3) intensification to improve the quality of existing solutions. The Algorithm 1 presents the generic structure of a metaheuristic.

Algorithm 1: General Metaheuristic

Input: ES, SS, GS, US, $N \geq 1$, $\lambda \leq N$, θ

Output: s_b (Best Solution)

- 1 Initialize N solutions S_t randomly
 - 2 Evaluate each member of S_t with an evaluation strategy (ES)
 - 3 Mark the best solution of S_t as s_b
 - 4 **repeat**
 - 5 Choose λ solutions from S_t using a selection strategy (SS)
 - 6 Create θ new solutions (set S'_t) using a generation strategy (GS)
 - 7 Evaluate new solutions (S'_t) with the same ES
 - 8 Create a new population S_t using an update strategy (US) by selecting N solutions from the set $S_t \cup S'_t$
 - 9 Mark the best solution of S_t as s_b
 - 10 **until** reach stop criteria;
 - 11 **return** s_b ;
-

Algorithm 1 resembles a genetic algorithm, however, to represent non-population based algorithms, such as Simulated Annealing, some adaptations are necessary, mainly in relation to the strategies used. The great advantage of using meta-heuristics as optimization algorithms is the level of abstraction that this solution has. For instance, in Algorithm 1 each type of strategy must be adapted based on the characteristics of the problem, that is, it is a malleable solution whose effort will depend on the objectives of the researcher.

The framework allows the researcher to implement his own optimization algorithm as long as it respects the input and output formats of the execution. The interface developed for the implementation of an optimizer is divided into three levels of specialization. At the most abstract level, the user simply implements a function that returns a similarity matrix M containing the candidate matches and their confidence rates. At the intermediate level, the user has the inclusion of a Model object that delivers the methods of accessing the data of the ontologies. At the third level, the user can include in his optimization algorithm an objective function created based on a fitness interface. Like the optimization algorithms, the framework provide a generic interface allowing the researcher to implement his own objective function or use one (or more) of those available. To implement a new objective function, the interface requires the researcher

to define the fitness calculation method and other simple characteristics regarding the function, for example, if the objective is to maximize or minimize.

To exemplify the use, three optimization approaches are available in the framework, a genetic algorithm, a prey-predator algorithm and a greedy randomized adaptive search procedure (GRASP). Each of the three example algorithms is implemented at the third level of specialization, that is, the algorithms implement a function that returns the similarity matrix M , use the Model object and use an objective function based on the Fitness interface.

The implementation of GRASP has two parts: (1) a constructive step, where a heuristic builds the initial solution, and (2) a iterative local search process to find better solutions in the neighborhood of the initial solution. In the first step, N solutions are randomly created, evaluated according to an evaluation strategy (ES), and inserted in a set of solutions S_t . The best S_t solution is submitted to a local search process. The local search is a process of refinement of a solution in which it seeks to find better solutions in its neighborhood, where the neighborhood is defined by the application of changes in the solution structure. In the proposed algorithm, the chosen neighborhood was the application of the addition and subtraction operators of a granularity factor G for each element of the vector that encodes the solution, having used the strategy of the best improvement in updating the solution. At the end of the process, the best solution found after the local search is compared with the best current solution in order to choose the winner. Steps (1) and (2) can be performed several times according to an execution parameter. A pseudocode of the GRASP algorithm is presented in Algorithm 2.

Algorithm 2: GRASP

Input: ES, G , $N \geq 1$, θ
Output: s_b (Best Solution)

- 1 **repeat**
- 2 Initialize N solutions S_t randomly
- 3 Evaluate each member of S_t with an evaluation strategy (ES)
- 4 Mark the best solution of S_t as sl_b
- 5 **repeat**
- 6 Create θ new neighboring solutions (set S'_t) using sl_b and a granularity G
- 7 Evaluate new solutions (S'_t) with the same ES
- 8 Mark the best solution of S'_t as sl_b if it is better than the current sl_b
- 9 **until** reach stop criteria;
- 10 Update s_b if sl_b is a better solution
- 11 **until** reach stop criteria;
- 12 **return** s_b ;

The main elements of a genetic algorithm (GA) are the selection, crossing, and mutation operators. In this work, the GA implementation uses a roulette method in the selection operator. After the selection, the single crossover point is applied in genes. The mutation is a unary operator that adds parent independent genetic characteristics in some new chromosomes.

Inductive mutation (POLI et al., 2008) is applied to a new child chromosome with probability ρ_m . A mutated chromosome gene, denoted g , is randomly selected. Next, a random value $x \in \{-2, -1, 1, 2\}$ is selected. A new value $g' = g + xG$ is assigned to g if $g' \geq 0$, where G is the granularity value. Otherwise, the process repeats for other randomly chosen chromosome.

A solution improvement technique is used to search a winning neighborhood for better solutions. A local search is used in the improvement process. The process is the same used by the GRASP when searching for neighboring solutions. Improvement process is performed each t generations. Periodicity t usually is set as 10% of total number of generations i . It was developed an elitist genetic algorithm, in which the best solutions are moved to the next generation. Each new generation preserves the top t_m best individuals from the previous generation P_{i-1} . Improvement solution phase tries to find a better solution C_z in the neighborhood of w . If a solution C_z is found, then C_z is assigned to w . A mortality operator is applied to avoid persisting solutions for several generations, such as super-individuals. Individuals older than a given age m are not allowed in the next generation. A short version of the algorithm is listed in Algorithm 3.

Algorithm 3: Genetic algorithm

Input: System parameters

Output: *Winner* (best solution)

```

1 Creates the first population  $P$ ;
2  $Winner \leftarrow best\_solution(P)$ ;
3 while stop criteria is not reached do
4   | Selects pairs to crossover;
5   | foreach pair  $(c, c')$  do
6   |   | Crossover  $(c, c')$ , with selection probability  $\rho_r$  and mutation probability  $\rho_m$ ;
7   | end foreach
8   | Explores the neighborhood of Winner with periodicity  $t$ ;
9   |  $P \leftarrow update\_population(P)$ ;
10  |  $Winner \leftarrow best\_solution(P)$ ;
11 end while
12 return Winner;

```

Several objective functions can be found in the literature, in this research we classified them into supervised and unsupervised approaches, in order to simplify the understanding of how these functions can work. A supervised objective function requires a reference alignment to evaluate the quality of a solution, while the unsupervised function works independently. The following are the objective functions available in the framework as well as their classification.

Let A^* be the ideal alignment and R the output alignment returned by the framework, the Precision score (Equation 3.1) calculates which percentage of returned alignments are correct, meanwhile the Recall score (Equation 3.2) calculates how much the returned alignments represent from the expected total. The two measures are opposed, because for Precision the smaller the size of R , the better the metric evaluation. In the recall, the larger the size of the response set R , the better the result. To balance the two, the F -measure score (Equation 3.3) is

computed as a harmonic mean between Precision and Recall. Precision, Recall and F-measure are supervised fitness functions, as their calculation depends heavily on the reference alignment (A^*) to assess the quality of the solution.

$$Precision = \frac{|R \cap A^*|}{|R|} \in [0, 1] \quad (3.1)$$

$$Recall = \frac{|R \cap A^*|}{|A^*|} \in [0, 1] \quad (3.2)$$

$$F = 2 * \frac{Precision * Recall}{Precision + Recall} \in [0, 1] \quad (3.3)$$

Some authors work with a reduced set of reference correspondences. In this case, the framework has a fitness function based on the construction of a linear system created from the reference alignment, where the closer the solution is to solving the linear system, the better its fitness value. Consider S a set of known matches of equivalence. The set S is formed by correspondence tuples $(e_{1i}, e_{2i}, \equiv, s_i)$, where e_{1i} and e_{2i} are distinct ontology entities, \equiv denotes the equivalence type relationship and s_i is the known similarity, reported by the ontology engineer, between e_{1i} and e_{2i} . Let f be a similarity measure composed of the weighted sum of other functions, by applying the function f to e_{1i} and e_{2i} , we expect to find the value s_i , that is, $f(e_{1i}, e_{2i}) = s_i$. As an example, consider the set $S' = (e_{11}, e_{21}, \equiv, 1), (e_{12}, e_{22}, \equiv, 1), (e_{13}, e_{23}, \equiv, 1)$, with all matches having similarity equal to 1. Consider a function $\bar{f}'(e_{11}, e_{21}) = g_1(e_{11}, e_{21})w_1 + g_2(e_{11}, e_{21})w_2 + g_3(e_{11}, e_{21})w_3$, where g_l represents the similarity value defined by the l^{th} function which are constants of the problem and w_l represents the weight assigned to the l^{th} function. For each known alignment provided in the input, it is possible to construct a linear system:

$$\begin{aligned} \bar{f}'(e_{11}, e_{21}) &= s_1 \therefore g_1(e_{11}, e_{21})w_1 + g_2(e_{11}, e_{21})w_2 + g_3(e_{11}, e_{21})w_3 = 1 \\ \bar{f}'(e_{12}, e_{22}) &= s_2 \therefore g_1(e_{12}, e_{22})w_1 + g_2(e_{12}, e_{22})w_2 + g_3(e_{12}, e_{22})w_3 = 1 \\ \bar{f}'(e_{13}, e_{23}) &= s_3 \therefore g_1(e_{13}, e_{23})w_1 + g_2(e_{13}, e_{23})w_2 + g_3(e_{13}, e_{23})w_3 = 1 \end{aligned} \quad (3.4)$$

In the objective function based on the linear system, the closer a solution is to solving the system, the better the quality of that solution. The formalization of this minimization function is presented in Equation 3.5. The problem is that in some scenarios there is no reference alignment and researchers are forced to look for alternative solutions.

$$Min(F_{linear}) = \sum_{i=1}^{|s|} 1 - s_i \in [0, |s|] \quad (3.5)$$

Match Coverage and Match Ratio (both used in (XUE; WANG, 2015b), (XUE; WANG, 2015c)) are unsupervised objective functions used to approximate the values of Recall and

Precision. Match coverage score (Equation 3.6) evaluates the fraction of entities which exist in at least one correspondence in the resulting alignment when compared to the total number of entities in the ontology, it is used as a substitute for recall. Let E_{O_1} and E_{O_2} be the set of all entities of ontology O_1 and O_2 respectively, and $E_{O_1-Match}$ and $E_{O_2-Match}$ be the set of all matched entities of O_1 and O_2 respectively. Match Coverage is defined as:

$$MatchCoverage = \frac{|E_{O_1-Match}| + |E_{O_2-Match}|}{|E_{O_1}| + |E_{O_2}|} \in [0, 1] \quad (3.6)$$

While MatchCoverage simulates Recall, MatchRatio score simulates Precision's behavior. MatchRatio (Equation 3.7) evaluates the fraction of entities which exist in at least one correspondence in the resulting alignment when compared to the set of correspondences in the resulting alignment. Let $Corr_{O_1-O_2}$ be the set of correspondences in the resulting alignment, MatchRatio is defined as:

$$MatchRatio = \frac{|E_{O_1-Match}| + |E_{O_2-Match}|}{|2 * Corr_{O_1-O_2}|} \in [0, 1] \quad (3.7)$$

In the same manner that F-measure combines Precision and Recall, MatchFmeasure combines MatchRatio and MatchCoverage. MatchFmeasure is defined as:

$$MatchFmeasure = 2 * \frac{MatchCoverage * MatchRatio}{MatchCoverage + MatchRatio} \quad (3.8)$$

The last unsupervised objective function available in the framework is based on the confidence attributed to each correspondence present in the alignment. Suppose an alignment a containing $|a|$ correspondences, the sum of confidences function is computed as:

$$ConfidenceSum = \sum_{it=1}^{|a|} \eta(c_{it}) \quad (3.9)$$

where $\eta(c_{it})$ represents the confidence value for the it^{th} match (correspondence) in a . The premise behind Equation 3.9 is that the greater the sum of the confidence, the more chances are there that the alignment is correct. Altogether there are four unsupervised objective functions (MatchCoverage, MatchRatio, MatchFmeasure and ConfidenceSum) and four supervised functions (linear system, Precision, Recall and F-measure) that depend on the reference alignments provided at the beginning of the experiment.

The output of the optimization module is a similarity matrix that contains candidate correspondences. Let $E_{1,i}$ be the i^{th} entity of O_1 and $E_{2,j}$ be the j^{th} entity of O_2 , where $i \in \{1, |O_1|\}$ and $j \in \{1, |O_2|\}$. Each row of the matrix M has a tuple $(E_{1,i}, E_{2,j}, \eta_{i,j})$ where $\eta_{i,j}$ is the confidence value for the candidate match $(E_{1,i}, E_{2,j})$.

3.4.4 Correspondence Selection

The correspondence selection module is responsible for selecting the candidate matches of matrix M that are most likely to be a correct match. The algorithms used in this step vary according to the researcher's objective. The size of the ontologies also influences the choice of correspondence selection algorithms since the weights adjustment process of the optimizer is costly for large-scale ontologies. What is observed in the literature is that authors who work with large-scale ontologies allocate effort in the stage of correspondence selection, making use of more complex algorithms such as memetics (XUE; LIU, 2017a), NSGA-III (XUE; LU; CHEN, 2019) and others (XUE et al., 2018a; XUE; CHEN, 2019c). Meanwhile authors who work with conventional ontologies and focus on the optimization process use simpler selection algorithms such as greedy heuristics and other deterministic algorithms (SOUZA; SIQUEIRA; NUNES, 2019; ACAMPORA et al., 2013a). This framework allows the researcher to implement their own selection algorithms through a standard interface, whether they are specific heuristics or meta-heuristics. To propose a new solution, the researcher needs to implement a method that uses the M similarity matrix as an input and returns a set of matches that represent the final alignment.

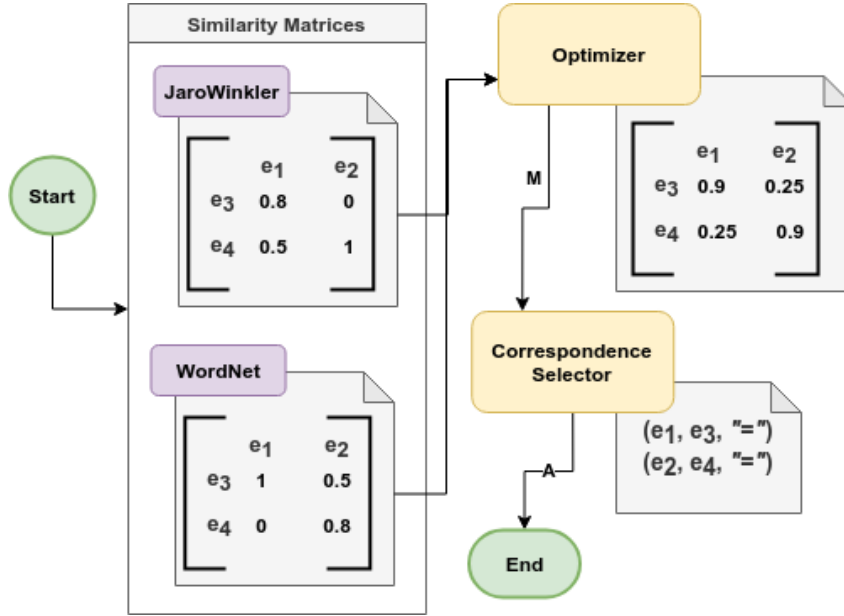
To exemplify this process, Figure 12 shows the data flow starting from the confidence value calculated by each matcher for each candidate correspondence. If e_1 and e_2 are entities in the source ontology and e_3 and e_4 are entities in the target ontology, the experiment optimizer combines the confidence value while the match selector chooses the most reliable pairs. This framework provides three selection algorithms as examples. The first algorithm is a iterative greedy algorithm that seeks, at each iteration, the highest $\eta_{i,j}$ and selects the most reliable match. The second algorithm is based on the Hungarian method and details are presented in (KUHN, 1955). The third algorithm presents a Simulated Annealing metaheuristic implementation that, unlike the others, makes use of a structure to represent a solution based on correspondences. This structure is available for consumption in other algorithms and consists of a list of objects, each one containing the information about a single correspondence.

The algorithms available in the framework work with cardinality rates (1:1), which means, that each entity can only be aligned once with another entity. Therefore, since two entities are chosen as a match, all other candidate matches that have one of these entities must be removed from matrix M . Other cardinalities can be added. The output format of this module is an alignment (A) that contains the most reliable correspondences of M .

3.4.5 Statistics Generator

The statistics module is part of the experiment evaluation activity. The module uses as input the data captured by the provenance model and calculates the main metrics that evaluate the quality of the alignment such as Precision, Recall and F-measure. To facilitate comparison between experiments, a set of correlation measures was included. The Jaccard coefficient detailed

Figure 12 – Data Flow of the meta-matching process, showing the confidence calculated by each similarity measure for each pair of entities, the aggregated confidence calculated by the Optimizer, and the alignment chosen by the Correspondence Selector module



Source: created by the author.

in Equation 3.10, has been used by recent ontology matching studies (HERTLING; PORTISCH; PAULHEIM, 2019) as a measure of correlation between sets. It measures the similarity between two alignments a_1 and a_2 through the similarity between the correspondences that make up the alignments.

In addition to the Jaccard coefficient, this work includes the overlap coefficient (Equation 3.11) and the Dice coefficient (Equation 3.12). The first one measures the overlap between two finite sets, it is defined as the size of the intersection divided by the smaller of the size of the two sets. If set a_1 is a subset of a_2 or the converse then the overlap coefficient is equal to 1. The second one can be considered a semimetric (a metric that doesn't necessarily satisfy the triangular inequality) version of the Jaccard, and it is defined as twice the intersection of the sets divided by the sum of the number of elements of each.

$$Jaccard(a_1, a_2) = \frac{|corr(a_1) \cap corr(a_2)|}{|corr(a_1) \cup corr(a_2)|} \in [0, 1] \quad (3.10)$$

$$Overlap(a_1, a_2) = \frac{|corr(a_1) \cap corr(a_2)|}{\min(|corr(a_1)|, |corr(a_2)|)} \in [0, 1] \quad (3.11)$$

$$Dice(a_1, a_2) = \frac{2 * |corr(a_1) \cap corr(a_2)|}{|corr(a_1)| + |corr(a_2)|} \in [0, 1] \quad (3.12)$$

The qualitative and quantitative measures are complementary. If, on one hand, the correlation measures are not enough to evaluate the quality of an experiment alone, the use of

only quantitative metrics such as Precision, Recall and F-measure is not enough to assist the researcher in identifying the differences between two correspondences of distinct experiments. This framework included measures to evaluate experiments quantitatively (Precision, Recall, F-measure, Fallout, Overall), statistically (mean and standard deviation) and qualitatively (Jaccard, Dice, Overlap, and a table that classifies each correspondence of the alignment into true positives and false positives).

3.4.6 Provenance Collector

The execution of scientific experiments modeled as workflows produces results that need validation and that can be reproduced by third parties. For this, it is necessary to consult the history of data transformations from its origin to the results generated in the execution of a scientific workflow. That record is known as provenance. In order to provide more reliability to users or other programs that consume this framework, a module was developed to store relevant data generated throughout the experiment. Various data are consumed and generated within the meta-matching process. To build a provenance model it is necessary to define which data and processes of each stage will be stored. For this, the main information related to each stage of the ontology meta-matching experiment was raised:

- Model
 - What are the source and target ontologies?
 - Which is the training alignment set (if any)?
 - Which is the reference alignment set?
- Optimizer
 - Which meta-heuristic was used in the experiment?
 - What parameters were used by the meta-heuristic (population size, cooling rate, mutation rate, seed, etc.)?
 - What is the objective function?
 - What are the combination parameters found by the meta-heuristic for each matcher?
- Correspondence Selector
 - Which is the similarity matrix for all pairs of entities?
 - Which technique will be used to extract the final alignment from the M matrix and what are its parameters (if any)?
- Evaluation

- What are the evaluation metrics (for instance, precision, recall and f-measure values, processing time, etc) resulting from the experiment?

The provenance can be classified as prospective, when related to the workflow specification, or retrospective, when associated with the workflow execution (FREIRE et al., 2008). As the stages of the meta-matching process are fixed and described by a predetermined workflow, the prospective provenance is already captured. To store the retrospective provenance, data models can be used, based on the PROV model, which proposes a generic provenance representation, serializations, and definitions to support the interchange of provenance information on the Web (LEBO et al., 2013). PROV¹ is a set of recommended standards by World Wide Web Consortium². The PROV Ontology³ (PROV-O) expresses the PROV Data Model using Ontology Web Language 2 (OWL2). It provides a set of classes, properties, and restrictions that can be used to represent and interchange provenance information generated in different systems and under different contexts. It can also be specialized to create new classes and properties to model provenance information for different applications and domains. The PROV-O (LEBO et al., 2013), whose representation is shown in Figure 13, represents the basic structure to capture the flow of activities of an experiment. The initial three classes (Activity, Entity and Agent) are the basis for creating new domain-specific classes. An **Entity** captures a thing in the world (in a particular state). The entity *was derived from* some other entity and *was generated by* an **Activity** that *used* other entities. An **Agent** (e.g. a person or software execution) *was associated with* the activity, and the entity that *was generated by* the activity *was attributed to* that agent.

Figure 14 shows the main components of the provenance model for ontology meta-matching built based on the PROV-O model. Purple rectangles represent an activity in the matching process, orange pentagons represent an agent within the experiment, and lastly, yellow circles represent an entity. Each activity shown in Figure 14 describes a step in the framework's execution flow in which agents define how entities will be consumed by the activities. A sample of the provenance file is shown in Figure 15.

The data present in the file in Figure 15 guarantees to track the information used in each step of the experiment, allowing its reproduction, evaluation and better understanding. The modular architecture allows the experiment to be resumed at a specific point, which allows the researcher to evaluate the behavior of a specific desired algorithm. The provenance data are stored using RDF/Turtle, which makes easy its use by other tools.

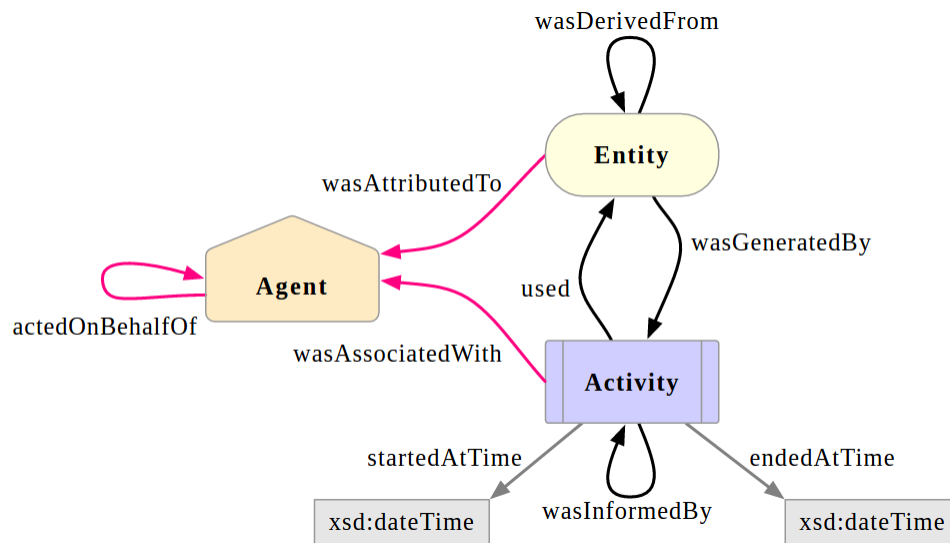
An overview of the architecture of this framework can be seen through Figure 11, highlighting the communication of each module and what information is used as input and output.

¹ <https://www.w3.org/ns/prov>

² <https://www.w3.org/>

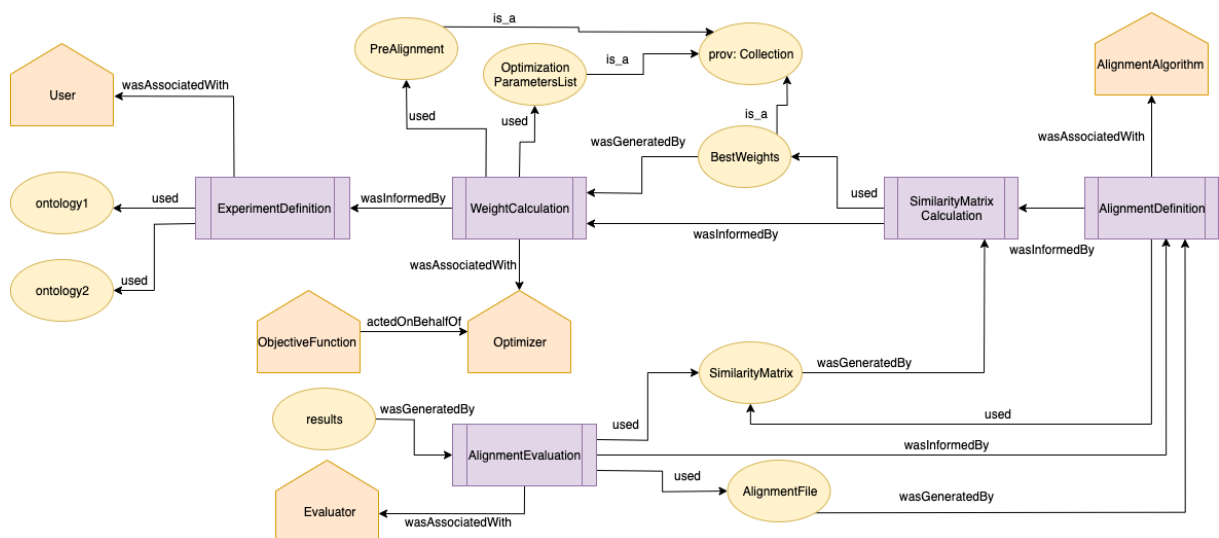
³ <https://www.w3.org/TR/2013/REC-prov-o-20130430/>

Figure 13 – Starting classes of the PROV-O ontology



Source: (LEBO et al., 2013)

Figure 14 – Provenance Model for Ontology Meta-Matching. The model starts by capturing the definition of the experiment, followed by the calculation of weights, calculation of the similarity matrix, definition of the alignment and, finally, the alignment evaluation.



Source: created by the author.

3.5 EVALUATION

This section is dedicated to reporting the results obtained through the use of the framework as a tool to support the development and scientific experimentation of ontology meta-matching approaches. Our experiments were carried out using a dataset provided by the Ontology Alignment Evaluation Initiative⁴ (OAEI).

⁴ <http://oaei.ontologymatching.org/>

Figure 15 – Provenance RDF-based Output File

```

:AlignmentAlgorithm_1578511383523
  a prov:Agent;
  prov:value      "similarity.combination.HungaroCombination";
  rdfs:label      "AlignmentAlgorithm";
.

:WeightCalculation_1578511383523
  a prov:Activity;
  prov:wasAssociatedWith :Optimizer_1578511383523;
  prov:used             :PreAlignment_1578511383523;
  prov:used             :OptimizationParametersList_1578511383523;
  rdfs:label            "WeightCalculation";
.

:ConceptNameSimilarity_DamerauLevenshteinEditDistance_1578511383523
  a prov:Agent;
  prov:value           0.035^^xsd:decimal;
  rdfs:label           "ConceptNameSimilarity_DamerauLevenshteinEditDistance";
.

:ConceptNameSimilarity_StoilosMetric_1578511383523
  a prov:Agent;
  prov:value           0.865^^xsd:decimal;
  rdfs:label           "ConceptNameSimilarity_StoilosMetric";
.

:DirectIndividualByNameSimilarity_DamerauLevenshteinEditDistance_1578511383523
  a prov:Agent;
  prov:value           0.099999994^^xsd:decimal;
  rdfs:label

```

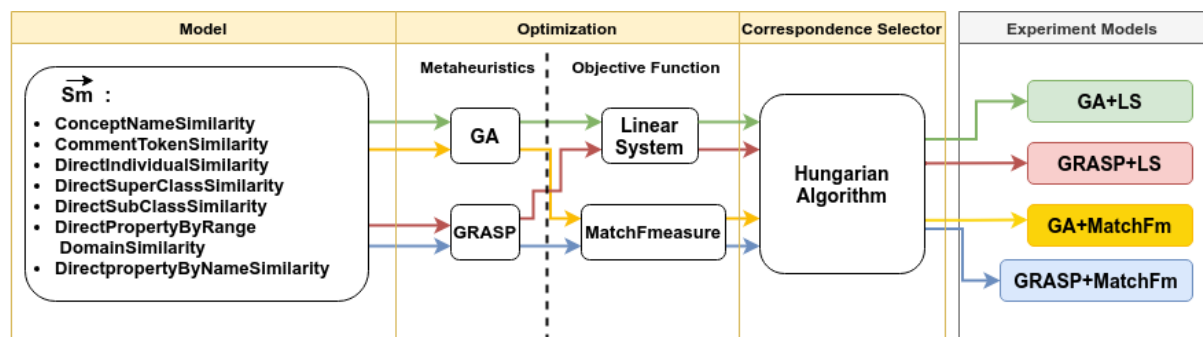
Source: created by the author.

Among the benchmarks made available by the OAEI, the biblio benchmark is the one that better suits the ontology meta-matching problem. Each test case presented in the biblio benchmark consists of the matching between a synthetic source ontology and a target ontology. The source ontology is the same in each test case, while the target ontology varies according to the test number. In test ranges 1xx and 2xx, the target ontology is created systematically from the source ontology where, at each test, part of the ontology information is discarded/modified in order to evaluate how the algorithm behave when this information is lacking. In range 3xx, the matching between the source ontology and real ontologies in the domain of bibliographic references is verified. This base test is suitable for meta-matching because what is expected from this type of approach is an adaptive behavior that rewards the most efficient solutions to maximize the quality of the result in different scenarios.

In the literature, there are several ontology meta-matching approaches using distinct similarity measures. When an approach is evaluated, the authors usually compares the accuracy and time cost. The meta-matching process has several steps where different algorithms can be applied. Comparing only these two aspects of the result makes it difficult to analyze which algorithm is more significant in the quality of the solution. The information about the individual contribution and characteristics of the matcher internal algorithms can be useful to the researcher when building and evolving a meta-matcher. In the conducted experiment, we sought to evaluate the behavior and impact of different objective functions applied to different optimization algorithms. Predetermined algorithms were fixed in some specific processing steps (definition of similarity measures and correspondences selection algorithm) while we ran experiments with different optimization algorithms and objective functions. It was used the Hungarian algorithm

(KUHN, 1955) in the selection stage. Seven basic similarity measures for determining the equivalence of two ontology entities were used in the experiments. The measures compare concept names, individuals, property names, property restrictions, concept metadata (comments) and the direct classes and subclasses of the concept. In this study, the framework was used to evaluate the behavior of two objective functions, a semi-supervised objective function based on linear system and the unsupervised objective function, MatchFmeasure. A genetic algorithm (GA) and a GRASP implementation were selected to adjust the weight of the similarity measures. Figure 16 presents the configuration of the experiments.

Figure 16 – Configurations created for evaluation. Each yellow container represents a processing module, the white balloons represent the algorithms used in each module. The name of the configurations used in the experiments are in the grey rectangle.



Source: created by the author.

The dataset was divided into subset tests. The subset 1xx is simpler than the others, as it contains only generalization or language restriction misrepresentations. The subset 3xx contains all tests with real ontologies. Tests 2xx, which contains misrepresentations related to hierarchy, instances, nomenclature, classes and others, was subdivided into two subsets. The subset 2xx-1 contains all tests that have one or two misrepresentations, while subset 2xx-2 contains test cases that have more than two misrepresentations. All tests were conducted by an Ubuntu 14.04.5 LTS machine with Intel Xeon CPU E5-2650 v2 @ 2.60GHz and 16GB RAM. The average f-measure grouped by subsets is presented in Table 16, the parameters of GA and GRASP were defined empirically and are presented, respectively, in Tables 14 and 15.

Table 14 – Genetic Algorithm Configuration

Parameter	Value
Population	100
Generations	100
Mutation Rate	0.1
Crossover Rate	0.8
Granularity	0.005
Selection Rate	0.5

Source: created by the author.

Table 15 – GRASP Configuration

Parameter	Value
Constructive phase: total solutions	20
Total iterations in local search	50
Repetitions number	03
Granularity	0.005

Source: created by the author.

Table 16 – Precision, Recall, and F-measure rates for the four experiment models

	#Test	Precision	Recall	F-measure
GA+LS	1xx	1.000	1.000	1.000
	2xx-1	0.934	0.957	0.942
	2xx-2	0.402	0.443	0.416
	3xx	0.604	0.613	0.607
GRASP+LS	1xx	1.000	1.000	1.000
	2xx-1	0.940	0.964	0.948
	2xx-2	0.358	0.398	0.372
	3xx	0.799	0.800	0.797
GA+MatchFm	1xx	1.000	1.000	1.000
	2xx-1	0.939	0.963	0.948
	2xx-2	0.353	0.395	0.367
	3xx	0.796	0.796	0.793
GRASP+MatchFm	1xx	1.000	1.000	1.000
	2xx-1	0.944	0.967	0.952
	2xx-2	0.407	0.452	0.423
	3xx	0.808	0.807	0.804

Source: created by the author.

All models obtained the maximum result in the subset 1xx and very close results in the subset 2xx-1 where the biggest difference was 0.006 between the GA+LS experiment and the GRASP+LS experiment, which makes it difficult to analyze the contribution of each algorithm individually. The subset 2xx-2 showed alternating results, with no pattern for contribution analysis. The most interesting behavior occurs in the subset 3xx, where the experiments that used GRASP or the MatchFmeasure function managed to achieve the best results in relation to f-measure, showing that both a meta-heuristic and a different objective function can explore more promising regions of the search space, making the meta-matcher able to make a leap of the same proportions in the final quality of the solution. The GRASP+MatchFm experiment suggests that GRASP and MatchFmeasure find different matches or enhance each other's results since their performance together surpasses others. Table 17 shows the Jaccard coefficient for the test number 301, which most closely resembles the average behavior of the 3xx range. It is possible to observe that the GRASP+LS and GA+MatchFm experiments found the same set of matches, that is, when used together, GRASP and MatchFmeasure find new sets of correspondences.

Table 17 also shows the Dice and the Overlap coefficients for the 301 test. The correlation rate is even higher than that found by the Jaccard coefficient. As all sets have the same number of correspondences, there are no variations in the denominator of the overlapping coefficient, showing that 11% of new correspondences were found when using GRASP and MatchFmeasure.

Table 17 – Jaccard, Overlap, and Dice Coefficients for test number 301

Jaccard #301	GRASP+LS	GA+MatchFm	GA+LS
GRASP+MatchFm	0.803	0.803	0.506
GRASP+LS	1.000	1.000	0.428
GA+MatchFm	1.000	1.000	0.428
Overlap #301	GRASP+LS	GA+MatchFm	GA+LS
GRASP+MatchFm	0.891	0.891	0.673
GRASP+LS	1.000	1.000	0.600
GA+MatchFm	1.000	1.000	0.600
Dice #301	GRASP+LS	GA+MatchFm	GA+LS
GRASP+MatchFm	0.891	0.891	0.673
GRASP+LS	1.000	1.000	0.600
GA+MatchFm	1.000	1.000	0.600

Source: created by the author.

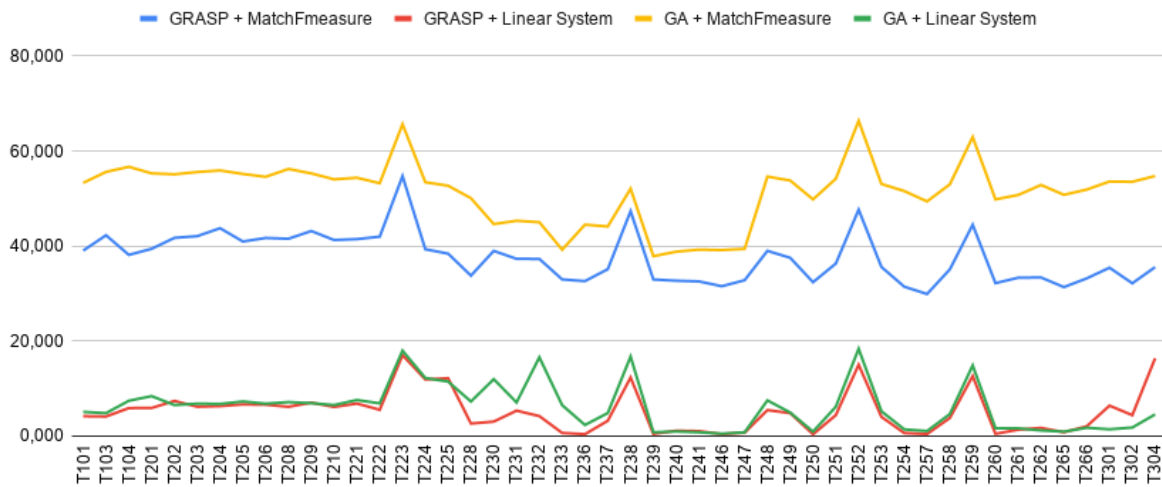
To verify the stability of the algorithms in finding solutions close to the optimum, the framework can be used to extract the mean and standard deviation of each experiment, as presented in Figure 18. The results clearly point to greater stability associated with the MatchFmeasure objective function (Figures 18c and 18d), which means that a smaller set of executions is necessary to achieve the optimal result of the model, while the Linear System (Figures 18a and 18b) presents a more dispersed behavior. However, as it is an objective function that takes into account only a small set of reference correspondences (from 3% to 4% of the total), the experiments showed that the Linear System has an average execution time in the order of $0.7 \cdot 10^{-4}$ seconds while MatchFmeasure spends an average of $0.6 \cdot 10^{-2}$ seconds to calculate, that is, the objective function based on the Linear System is approximately 86 times faster than MatchFmeasure. In terms of total execution time, Figure 17 demonstrates that all executions of experiments with the Linear System did not exceed twenty seconds of total time, while MatchFmeasure operated, in most cases, between thirty and sixty seconds, which highlights the time advantage related to the Linear System.

3.6 CONCLUSIONS

This article presented a freely available⁵ framework for scientific experimentation in ontology meta-matching. Considering the execution flow of popular OMM approaches, the framework was divided into independent modules to simplify the process of developing new

⁵ <<https://bitbucket.org/nicolasferranti/heuristicsontologymatching/src/master/>>

Figure 17 – Mean time in seconds for all experiments in all test cases



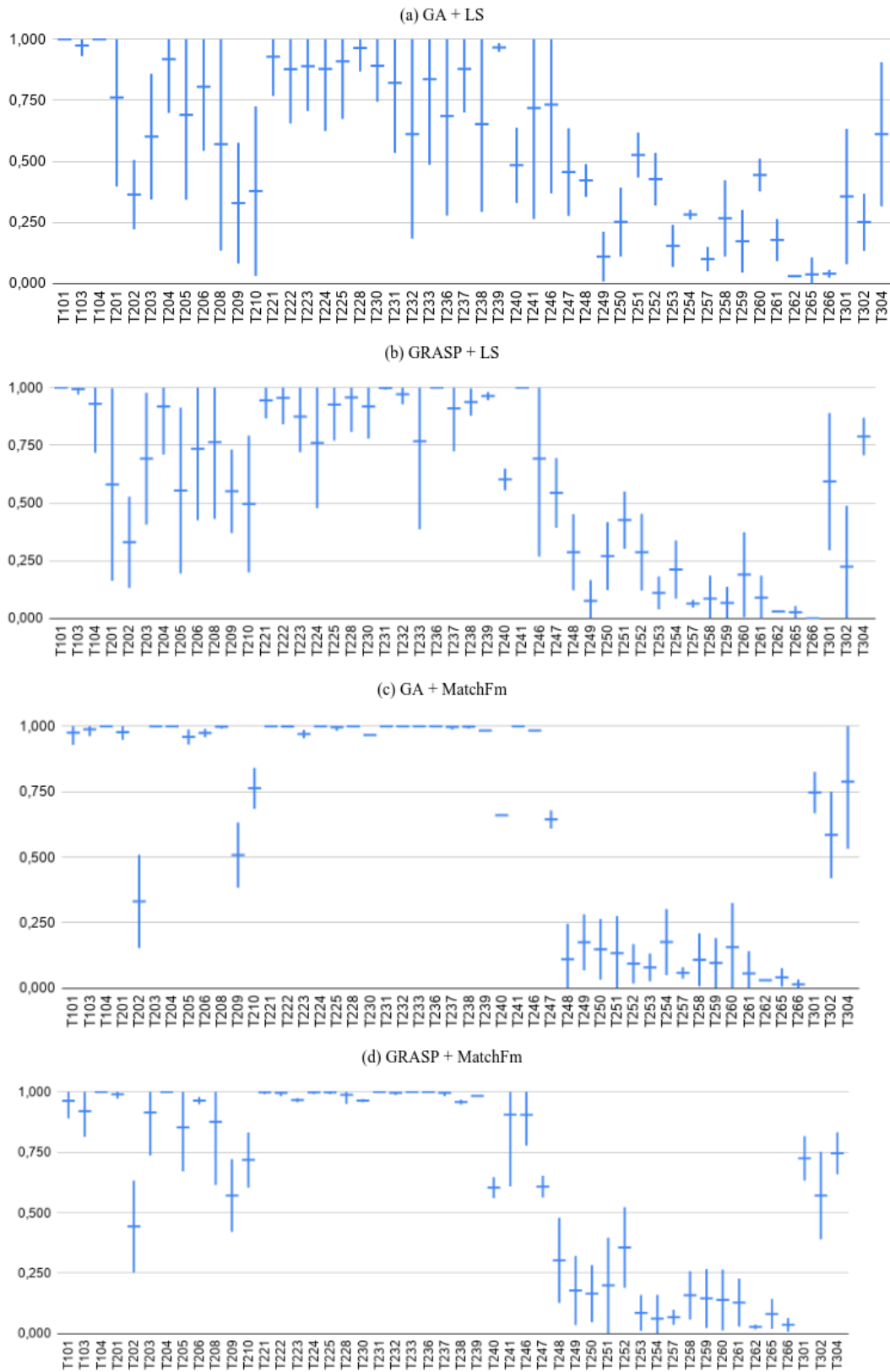
Source: created by the author.

OMM approaches. Statistical and provenance collection tools embedded in the framework assist the researcher to evaluate the experiments. The two main contributions of this work are: (1) the problem modeling in a simplified way that facilitates the coupling of new algorithms; (2) a data provenance model that collects the main information about the experiment. The experiments carried out to evaluate the framework point out important characteristics of each algorithm, characteristics that are relevant for decision making by researchers in the construction of meta-aligners that best suit their needs.

The evaluation section presented the behavior and the gain related to the use of two different objective functions, one based on MatchFmeasure and the other based on a Linear System. The results demonstrated that both a meta-heuristic and an objective function, when used correctly, can bring gains in the same proportion for the accuracy of an alignment. In addition, the quality stability of the solutions generated when using MatchFmeasure due to the fact that MatchFmeasure is applied over the entire set of entities while the Linear System is applied over a small reference alignment. In contrast, the Linear System has a low execution cost. It is worth noting that the metaheuristics can be used with any objective function, and the researcher can build his specific optimization algorithm in order to extract the best set of metaheuristic, objective function, and parameters.

The framework is useful to the community because it simplifies complex processing steps and delivers concise and well-defined information about the experiments. Once the source code is available, new researchers can contribute to the evolution of this tool. In future works, other stages of the framework (composition of similarity measures and correspondence selection algorithms) will be explored in order to study the impact of each algorithm in the matching process.

Figure 18 – Mean F-measure (horizontal line) and standard deviation of the mean F-measure samples (vertical line) for each experiment model for each test case



Source: created by the author.

4 MANUSCRIPT 3: EXPERIMENTAL RESULTS

Every year, new ontology matching approaches have been published to address the heterogeneity problem in ontologies. It is well known that no one is able to stand out from others in all aspects. An ontology meta-matcher combines different alignment techniques to explore various aspects of heterogeneity to avoid the alignment performance being restricted to some ontology characteristics. The meta-matching process consists of several stages of execution, and sometimes the contribution/cost of each algorithm is not clear when evaluating an approach. In this master's thesis, the main objective is to evaluate costs and contributions of algorithms commonly applied to the problem. This chapter is a full article that presents the evaluation of solutions commonly used in the literature in order to provide more knowledge about the ontology meta-matching problem. Results showed that, the more characteristics of the entities that can be captured by similarity measures set, the greater the accuracy of the model. It was also possible to observe the good performance and accuracy of local search based meta-heuristics when compared to global optimization meta-heuristics. Experiments with different objective functions have shown that semi-supervised methods can shorten the execution time of the experiment but, on the other hand, bring more instability to the result.

Section 4.4 of this chapter contains text describing the architecture and functioning of the framework that have already been presented in greater detail in Section 3.4.

4.1 INTRODUCTION

Ontologies are computational models for representing objects in a domain and their relationships. Due to its high representative capacity and its level of formalism, the ontology model, which is human-readable and machine-readable, has been used in several applications to model the underlying concepts in a formal manner (BANDROWSKI et al., 2016; HOEHNDORF; SCHOFIELD; GKOUTOS, 2015). The ontology-based modeling is subjective and expert-dependent; ontologies are built by people with distinct levels of expertise and goals. Therefore, concepts that describe the same type of object may be represented in different ways, both in syntax and structure, generating a problem of heterogeneity in data semantics. The heterogeneity among various data sources is the main obstacle to the semantic interoperability. Finding mappings between entities of different ontologies contributes to the interoperability of systems, facilitating the exchange of information and, consequently, reducing the problem of semantic heterogeneity (MOHAMMADI; HOFMAN; TAN, 2019).

Ontology matching techniques can be used to find semantic correspondences between ontology elements with less human effort. The ontology matching is a NP-hard optimization problem (KUREYCHIK; SEMENOVA, 2017) that can be addressed by several computational techniques. Due to the high heterogeneity of the ontologies, there is no technique that stands out from the others in all aspects (XUE; TANG, 2017). Ontology meta-matching (OMM)

approaches can be used in this scenario. An ontology meta-matcher approach uses methods to select the correct matching components to execute, and to adjust the multiple knobs (e.g., threshold, coefficients, weights, etc.) of the components that are part of the OMM process (XUE; PAN, 2018).

The literature suggests that the best results found in the meta-matching of ontologies are associated with the use of evolutionary algorithms (KUREYCHIK; SEMENOVA, 2017; XUE; TANG, 2017). The OMM process generally usually uses (1) multiple similarity measures, (2) a method of combining similarity measures, and (3) a method of selecting candidate matches, where evolutionary algorithms are commonly employed in at least one of these last two steps. Several evolutionary algorithms have been applied, and the most common are the Genetic Algorithm (GA) (GULIĆ; VRDOLJAK; PTIČEK, 2018), the Memetic Algorithm (RAMESH et al., 2016) and the Particle Swarm Optimization (PSO) (SEMENOVA; KUREYCHIK, 2016b).

Since OMM process has several steps, components, and other factors that may impact the result, comparing different approaches can be a challenging task. The authors normally use the absolute numbers of Precision, Recall, and F-measure, however, using these three measures is not enough to understand the pros and cons of each approach. This is because each approach is composed of different similarity measures, combination methods and search methods, What makes it not clear which researchers' idea was responsible for the improvement in the final result. An isolated analysis of the contribution/cost of each algorithm can help researchers and developers to choose computational solutions that best suit their usage scenario. The objective of this manuscript is to present a series of analyzes on the algorithms that are commonly used in each execution stage of an OMM tool. The analyzes were made through experiments that were designed to verify the behavior of each algorithm in isolation, that is, varying the algorithms that are used in a specific step but maintaining the algorithms of the other steps. In this way, the results are obtained according to the algorithms that were part of the object of study, since the other stages of execution do not vary from one experiment to another.

The rest of the manuscript is organized as follows: Section 4.2 presents related works that aimed to compare different approaches to OMM, as well as our view of the problem and how the experiment can be divided and executed. Section 4.3 presents part of the research methodology, introducing the hypotheses that were tested in the work. Section 4.4 presents the framework that was used to conduct the experiments, describing how it fits to accommodate the proposals of this research. Section 4.5 finalizes the methodology, explaining the experiment models built within the framework for each hypothesis to be tested. In section 4.6 we discuss the results and, finally, section 4.7 presents the conclusions and limitations of the work.

4.2 RELATED WORKS

Based on the workflow presented by some authors (SILVA; BAIÃO; REVOREDO, 2014; XUE; PAN, 2018), the ontology meta-matching task can be divided into steps. In the first step,

called by us as the Modeling step, authors usually describe which similarity measures are used to compare ontology entities. In the second step, the Optimization Step, a combination method is defined to aggregate multiple similarity measures. Several authors present approaches based on finding the best set of parameters to aggregate multiple similarity measures (XUE; PAN, 2018), which makes the second step a prolific research topic (OTERO-CERDEIRA; RODRÍGUEZ-MARTÍNEZ; GÓMEZ-RODRÍGUEZ, 2015; ACAMPORA; ISHIBUCHI; VITIELLO, 2014). After the optimization step establishes the final confidence value for each candidate match, it is necessary to select the correspondences that have greater probabilities of correctness. Thus, in the third step, Correspondence Selection, heuristic search methods are usually used to define the correspondence set (alignment) closest to the optimum (MOHAMMADI; HOFMAN; TAN, 2019).

Comparative studies on OMM approaches have evaluated the use of different meta-heuristics used in Optimization and Correspondence Selection steps. In (ACAMPORA; ISHIBUCHI; VITIELLO, 2014), the authors analyzed multi-objective ontology matching approaches, and aimed to evaluate the Optimization step. The authors make use of important metrics such as hypervolume, δ index, C metric, and others, which allow viewing the results from a statistical point of view. The main finding in (ACAMPORA; ISHIBUCHI; VITIELLO, 2014) is that the better performance of the Optimal Multi-Objective Particle Swarm Optimization (OMOPSO) over other famous multi-objective algorithms such as NSGA-II, MOEA/D, and SPEA2.

Seeking to analyze the application of local search meta-heuristics in the Correspondence Selection step, in (ACAMPORA; VITIELLO, 2020) the authors present a comparison on the TABU search, Simulated Annealing (SA) and other meta-heuristics. The experiments use the same set of similarity measures and integrate them in a weighted way through the OWA operator (Yager, 1988). In (ACAMPORA; VITIELLO, 2020), meta-heuristics are modeled to find the best candidate matches in one step after the aggregation of similarity measures, the goal is to find the alignment that minimizes the distance between entities (or maximizes the confidence value in the alignment). In addition to well-known metrics (Precision, Recall and F-measure), the authors (ACAMPORA; VITIELLO, 2020) compared the convergence rate of each of the metaheuristics. The convergence analysis in this case is facilitated by the modeling chosen by the authors, since the solution space is composed of candidate alignments, therefore, it is possible to compare the modifications of the final solution in each iteration. The authors highlighted the TABU search performance over other meta-heuristics.

MELT (HERTLING; PORTISCH; PAULHEIM, 2019) presents different tools to assess the similarity between the results of the matchers in order to promote different comparative analyses. MELT is a toolkit for ontology meta-matching whose components have been used individually in other researches (PAULHEIM, 2019). The toolkit was designed to evaluate ontology matching tools as well as visualizing matching results. Its main characteristics are associated with the visualization of results and the integration with the tools provided by OAEI,

such as SEALS and HOBBIT. The authors demonstrate the framework's evaluation methods by calculating similarity between the result of matchers who participated in Ontology Alignment Evaluation Initiative (OAEI) campaigns. The metric used in the article compares the similarity between correspondences found by different matchers, indicating the importance of qualitative analysis in the OMM problem.

Despite the advances in understanding the ontology meta-matching problem and approaches, there are some questions to be answered. New approaches have been published in the last years, but it is not clear whether the results are statistically significant and which new idea in the meta-matching process has resulted in better results. This occurs mainly due to the evaluation methodology adopted by the authors. In articles that compare different approaches, some authors (ACAMPORA; ISHIBUCHI; VITIELLO, 2014) compare different meta-heuristics in the Optimization step and others (ACAMPORA; VITIELLO, 2020) compare different meta-heuristics in the Correspondence Selection step, but when it comes to analyzing the impact of these algorithms on the solution, there is a gap. As the analysis is focused exclusively on meta-heuristics, there may be a false impression that the meta-heuristic was the main factor that contributed to the result. However, the OMM process is sequential and the result of one step directly affects the next step. Furthermore, the experiment is composed of similarity measures, objective functions, combination methods and other factors that may have a greater influence on the quality of the result or on the time spent to obtain it.

In some articles that present new proposals for OMM, the evaluation methods focus on accuracy results and do not explore relevant characteristics of the algorithms used. In (SEMENOVA; KUREYCHIK, 2016b), (BINIZ; AYACHI, 2018) and (XUE; REN, 2017), different OMM approaches are proposed, and the evaluations are made in comparing Precision, Recall and F-measure values. These three metrics are sufficient to validate whether the experiment is capable of finding solutions to the problem, but they are insufficient, when applied in completely different experiments, to answer which part of the approach is responsible for the results. In (SEMENOVA; KUREYCHIK, 2016b), the authors evaluate ontologies synthetically created and there is no comparison with other approaches in the literature. In (XUE; REN, 2017), the authors used the OAEI dataset and the results were compared to other approaches that were submitted to the same campaign using Precision, Recall and F-measure values. Analyzes that could bring more details of the results, however, are not easy to be done because there is a lack of information regarding the behavior of the algorithms that are executed in each processing step, such as the confidence value of each similarity measure, or the cost of performing an objective function.

Unlike MELT, this article presents comparisons using not only the correlation rate but the time spent and accuracy rates related to experiments with similar configurations to make clear the impact that a choice in the OMM workflow can cause to the results. Like (ACAMPORA; ISHIBUCHI; VITIELLO, 2014) and (ACAMPORA; VITIELLO, 2020), this study seeks to

find answers that help to understand the role that each algorithm plays in the construction of the solution. We analyze the contribution of different similarity measures, different objective functions, and comparisons between methods that are not exclusively based on meta-heuristics. These answers may allow researchers to better understand the statistical significance of each computational solution.

4.3 HYPOTHESIS DEFINITION

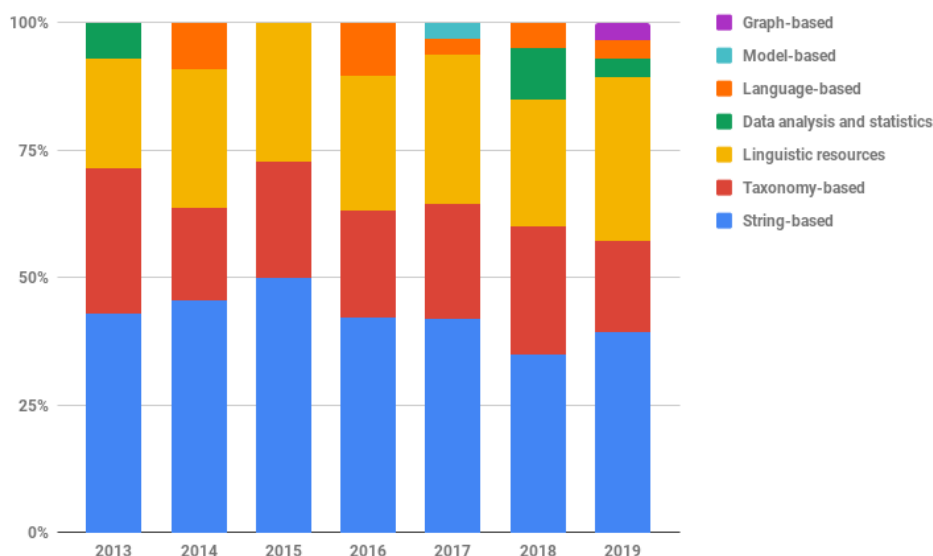
This section is dedicated to reporting the definition of each hypothesis tested in this work, as well as presenting the motivations and justifications behind each hypothesis. The first hypothesis concerns the use of multiple similarity measures and is defined as:

***H1** : To use String-based, Taxonomy-based and Linguistic Resources similarity measures is enough to create a resilient tool*

Although similarity measures represent the basis for starting the experiment, most articles do not include similarity measures in their evaluation methodologies; they only describe which measures were used and how they work, as in (XUE; WANG; REN, 2014; BINIZ; AYACHI, 2018; RAMESH et al., 2016). It is difficult to determine which measures contributed to a particular alignment. The **H1** hypothesis tests the ability of different similarity measures to find different matches, using similarity measures from three different groups. Similarity measures can be classified with respect to the type of input they work with. A well-known classification is presented in (SHVAIKO; EUZENAT, 2013). In our experiments, we evaluated measures from three groups: String-based, Taxonomy-based, and Linguistic Resources. The three groups were chosen because they represent the majority of similarity measures that have been used by researchers in recent years. Data on the most frequent groups of similarity measures can be seen in Figure 19. Figure 19 shows the percentage that each group represented in articles published in the last seven years, the groups are divided by the year of article publication and follow the classification of (SHVAIKO; EUZENAT, 2013).

Despite String-based, Taxonomy-based and Linguistic Resources are the most common types, there are other algorithms that can be used to assess the similarity between two objects, some of which may not have been used in OMM approaches. The knowledge domain represented by the ontology is a factor that can influence some authors to search for domain-specific similarity measures, for example, if the ontologies are in the health area it may be interesting to look for a similarity measure that uses a medical thesaurus. Domain-specific measures can enhance the results but end up being restricted to just one application scenario. The normalized compression distance (NCD) is a distance measure that works for any type of data and has been used in clustering applications (SARASA; GRANADOS; RODRIGUEZ, 2019). As far as we know, the NCD has not yet been tested on OMM tools. The **H2** hypothesis comes precisely to verify the applicability of the NCD.

Figure 19 – Percentage of each type of similarity measure per year



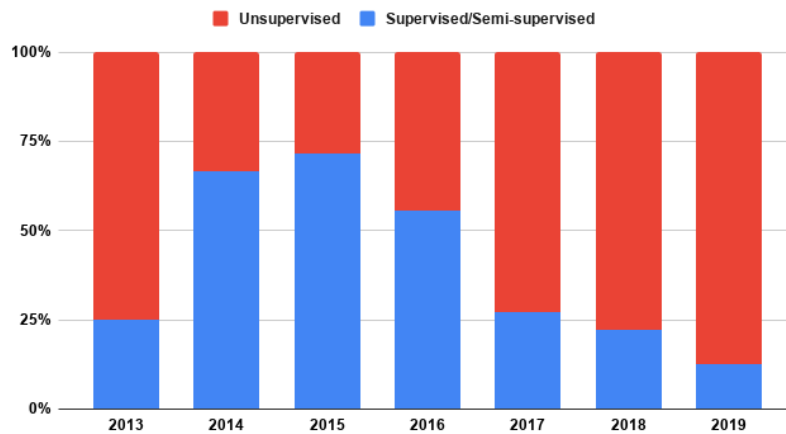
Source: created by the author.

H2 : The use of the normalized compression distance can reduce the error rate of a ontology meta-matcher

After defining the hypotheses about the similarity measures used in the Modeling step, the next hypotheses refer to the meta-heuristics and objective functions, dividing between those used in the Optimization step and the ones used in Correspondence Selection step. There are several works addressing the OMM problem using different meta-heuristics, such as Genetic Algorithm (GA) (SOUZA; SIQUEIRA; NUNES, 2019), Memetic Algorithm (XUE; TSAI; WANG, 2017) and Particle Swarm Optimization (PSO) (MARJIT, 2015). Each meta-heuristic can make use of one or more objective functions, which are responsible for guiding the meta-heuristic solution(s) in the search space.

In this work, we divide the objective functions into supervised/semi-supervised and unsupervised approaches. Supervised approaches require a full reference alignment to compute their value, semi-supervised approaches require a relatively small reference alignment, and unsupervised approaches works independently, with no reference alignment required. Figure 20 presents a comparison between the number of supervised/semi-supervised and unsupervised approaches in the literature, pointing to an increase in the number of unsupervised in recent years. In the Optimization step, both supervised/semi-supervised (SOUZA; SIQUEIRA; NUNES, 2019) and unsupervised (XUE; TSAI; WANG, 2017) approaches can be used. Both (SOUZA; SIQUEIRA; NUNES, 2019) and (XUE; TSAI; WANG, 2017) evaluate the research in the OAEI biblio benchmark, and the authors applied meta-heuristics in the optimization step, but with different objective functions. The evaluation of (SOUZA; SIQUEIRA; NUNES,

Figure 20 – Amount of Unsupervised vs Supervised/Semi-supervised approaches per year



Source: created by the author.

2019) and (XUE; TSAI; WANG, 2017) is done in terms of the F-measure, (XUE; TSAI; WANG, 2017) still includes comparisons of time and memory spent. Neither the articles that compare approaches (ACAMPORA; ISHIBUCHI; VITIELLO, 2014; ACAMPORA; VITIELLO, 2020), nor the articles that propose algorithms that fit in the Optimization step (SOUZA; SIQUEIRA; NUNES, 2019; XUE; TSAI; WANG, 2017), present evaluation proposals about different objective functions, making it difficult to understand the contribution of each objective function and, likewise, making it difficult to study new trends, such as the data presented in Figure 20. The **H3** hypothesis was built to evaluate two distinct objective functions, one supervised and one unsupervised.

H3 : *In the same experiment setup, an unsupervised approach can achieve results with quality close to that of a supervised one, however the time for reaching this solution is longer*

The last hypothesis was created based on the algorithms used in the Correspondence Selection step. There are several methods that can be used at this step (ESSAYEH; ABED, 2015), such as the Karp algorithm, Minimum Cost Flow (MCF), Hungarian Algorithm or meta-heuristics (MOHAMMADI; HOFMAN; TAN, 2019). Simpler deterministic algorithms like Hungarian optimize based only on the confidence value assigned to each relationship, while a meta-heuristic can evaluate other aspects such as the number of candidate matches for each entity. In (ACAMPORA; VITIELLO, 2020), the authors compare different local search meta-heuristics that use the same objective function to search for optimal alignment. The evaluation is conducted using the F-measure score and the solution convergence rate. In (MOHAMMADI; HOFMAN; TAN, 2019), a tool named SANOM is proposed, and the authors use SA to search for optimal alignment. The solution used in SANOM fits in the Correspondence Selection step described in this work. SANOM was evaluated by numerous tracks from OAEI, demonstrating the versatility

of the tool. The authors present analyzes of time consumption, Precision, Recall, F-measure and other metrics.

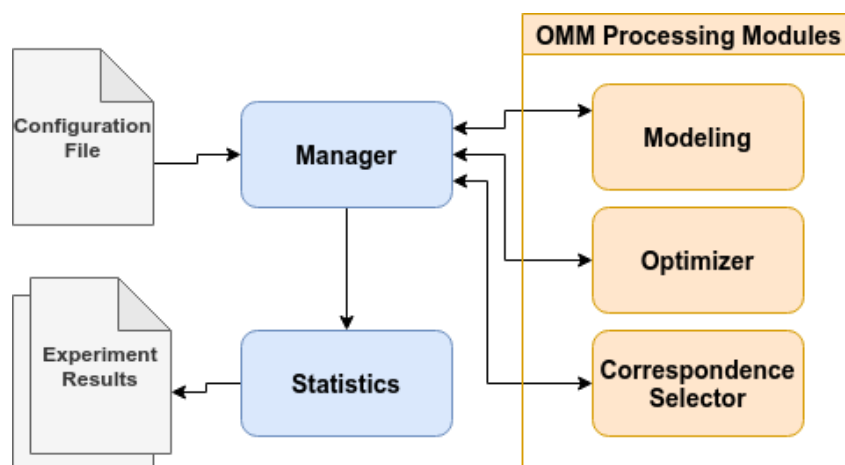
Both works (ACAMPORA; VITIELLO, 2020; MOHAMMADI; HOFMAN; TAN, 2019) present robust solutions for the correspondence selection step, however some authors use simpler methods and still are able to find good solutions (SOUZA; SIQUEIRA; NUNES, 2019; ESSAYEH; ABED, 2015). As the evaluation of (ACAMPORA; VITIELLO, 2020) is carried out by exchanging one meta-heuristics for another and the SANOM evaluation is conducted comparing the final result of the experiment, there is no evidence of the effects of the Hungarian algorithm, since the articles focus on methods based on meta-heuristics. The **H4** hypothesis tests the cost/benefit of a correspondence selection algorithm based on meta-heuristics and another based on the Hungarian method.

***H4** : To use a meta-heuristics selection approach can cause new matches to be found in a time similar to the Hungarian algorithm.*

4.4 ONTOLOGY META-MATCHING EXPERIMENTATION FRAMEWORK

In this work, we make use of a framework to test ontology meta-matching approaches. It was built based on the division of the ontology meta-matching process into steps, where a set of computational solutions is available for each step. The framework covers the three main steps of the OMM process: the Modeling, Optimization, and Correspondence Selection steps (see Figure 21). Each step constitutes a system module that includes algorithms to evaluate similarity between entities, optimization algorithms for combining similarity functions and selection algorithms to choose matches. Each module operates independently having its input data and a predefined output format. This architecture allows new approaches to be implemented at specific process steps facilitating the construction of different experiment models.

Figure 21 – System Architecture



Source: created by the author.

In Figure 21, it is possible to observe the existence of a management module that controls the flow of execution of the experiment. This module is responsible for guaranteeing the execution and delivering the result of one execution step to the subsequent step. The processing modules are those that have different algorithms to perform the same OMM sub-task. Through changes in the configuration file, new experiment models can be created, processed and evaluated under different measures of a statistics module. This framework is freely available¹ for researchers to use and improve it and the following subsections describe in detail each of the processing modules.

4.4.1 Modeling

The modeling module is the first to be triggered by the manager module. This module has as input the files containing the pair of ontologies that will be aligned and the name of each similarity measure that will be used in the experiment. This framework was developed to process ontologies described in OWL or RDF. All this information is read and structured in computational objects that will be consumed in the following steps by means of the Jena API (MCBRIDE, 2002). The output consists of objects containing the ontologies data and a set of similarity measures.

The user can implement his own similarity measures as well as use the measures available in the tool. Each similarity measure (matcher) generates a matrix with $|e_1|$ rows and $|e_2|$ columns, where e_1 represents the set of entities from the source ontology and e_2 the set of entities from the target ontology. Each cell in the matrix represents the confidence value assigned by that matcher for a specific candidate match. To include new similarity measures, the user simply implements the interface that best suits the data that will be needed in the measure. For example, a String-based measure (EUZENAT; SHVAIKO, 2013) can be added using the *IEditDistance* interface because it consumes two strings, while functions such as Similarity Flooding (MELNIK; GARCIA-MOLINA; RAHM, 2002) must use the *ISimilarityFunction* interface because it is necessary to use other information from the concept. Table 18 shows some of the measures available and their classification according to (EUZENAT; SHVAIKO, 2013).

Table 18 – Similarity Measures

Name	Classification	Reference
Wu Palmer	Linguistic-based	(WU; PALMER, 1994)
Lesk	Linguistic-based	(BANERJEE; PEDERSEN, 2003)
Similarity Flooding	Taxonomy-based	(MELNIK; GARCIA-MOLINA; RAHM, 2002)
JaroWinkler	String-based	(WINKLER, 1999)
Levenshtein	String-based	(LEVENSHTAIN, 1966)
DamerauLevenshtein	String-based	(DAMERAU, 1964)

Source: created by the author.

¹ <<https://bitbucket.org/nicolasferranti/heuristicontologymatching/src/master/>>

4.4.2 Optimization

The optimization step is the experiment step responsible for combining multiple similarity measures and defining a confidence value for each pair of ontology entities. An optimizer can be defined as a tuple $\sigma = (o_1, o_2, Sm, HSA)$ where:

- o_1 and o_2 are the computational structures of the ontologies to be matched, represented by an instance of type `Model` that contains the methods to retrieve the pertinent information of the ontology data.
- Sm represents the similarity measures set. For each $sim \in Sm$, $sim(Entity_1, Entity_2) = Conf_{sim}$ where $Entity_1, Entity_2$ are entities from distinct ontologies and $Conf_{sim} \in [0, 1]$.
- HSA is an heuristic search algorithm.

To perform the optimization process, it is common in the literature to use metaheuristics such as Genetic (XUE; PAN, 2018), Memetic (XUE; CHEN, 2019b), PSO (SEMENOVA; KUREYCHIK, 2016b) algorithms and many other approaches as search algorithm. Evolutionary algorithms are becoming the state of the art in the search for solutions to this problem (XUE; LIU, 2017b). These search heuristics are usually used to find the best combination of similarity measures. Despite having inspiration in different phenomena, the sequence of operations of the meta-heuristics are similar, generally making use of (1) a phase of building one or more solutions, (2) diversification of solutions to explore the search space and (3) intensification to improve the quality of existing solutions. The great advantage of using meta-heuristics as optimization algorithms is the level of abstraction that this solution has, each type of strategy must be adapted based on the characteristics of the problem, that is, it is a malleable solution whose effort will depend on the objectives of the researcher.

The framework allows the researcher to implement his own optimization algorithm as long as it respects the input and output formats of the execution. The interface developed for the implementation of an optimizer is divided into three levels of specialization. At the most abstract level, the user simply implements a function that returns a similarity matrix M containing the candidate matches and their confidence rates. At the intermediate level, the user has the inclusion of a `Model` object that delivers the methods of accessing the data of the ontologies. At the third level, the user can include in his optimization algorithm an objective function created based on a fitness interface. As the optimization algorithms, the framework provide a generic interface allowing the researcher to implement his own objective function or use one (or more) of those available. To implement a new objective function, the interface requires the researcher to define the fitness calculation method and other simple characteristics regarding the function, for example, if the objective is to maximize or minimize.

To exemplify the use, three optimization approaches are available in the framework: a genetic algorithm (GA), a prey-predator algorithm (PPA) and a greedy randomized adaptive search procedure (GRASP). Each of the three algorithms was implemented at the third level of specialization, that is, the algorithms implement a function that returns the similarity matrix M , use the Model object and use an objective function based on the Fitness interface.

Table 19 – Objective Functions Available

Name	Classification	Reference
Precision	Supervised	(MANNING; RAGHAVAN; SCHÜTZE, 2008)
Recall	Supervised	(MANNING; RAGHAVAN; SCHÜTZE, 2008)
F-measure	Supervised	(MANNING; RAGHAVAN; SCHÜTZE, 2008)
Linear System Based	Semi-Supervised	(SOUZA; SIQUEIRA; NUNES, 2019)
Confidence Sum	Unsupervised	(ACAMPORA et al., 2013a)
MatchCoverage	Unsupervised	(XUE; LIU, 2017c)
MatchRatio	Unsupervised	(XUE; LIU, 2017c)
MatchFmeasure	Unsupervised	(XUE; LIU, 2017c)

Source: created by the author.

Several objective functions can be found in the literature. In this research we classified them into supervised and unsupervised approaches in order to simplify the understanding of how these functions can work. A supervised objective function requires a reference alignment to evaluate the quality of a solution, while the unsupervised function works independently. Table 19 presents the set of objective functions available in the framework, as well as the classification we propose for the type of approach and the reference where more details of the algorithm can be found. The objective function based on the linear system was classified as semi-supervised because it uses a relatively small reference alignment when compared to the others supervised. The objective functions guide the search performed by the meta-heuristic in the solution space. At the end of the execution, the meta-heuristic returns the parameters to aggregate multiple similarity measures. These parameters are used to build a unified similarity matrix which represents the Optimization module output. This similarity matrix contains all possible candidate correspondences and their similarity score. Let $E_{1,i}$ be the i_{th} entity of O_1 and $E_{2,j}$ be the j_{th} entity of O_2 , where $i \in \{1, |O_1|\}$ and $j \in \{1, |O_2|\}$. Each row of the matrix M has a tuple $(E_{1,i}, E_{2,j}, \eta_{i,j})$ where $\eta_{i,j}$ is the confidence value for the candidate match $(E_{1,i}, E_{2,j})$.

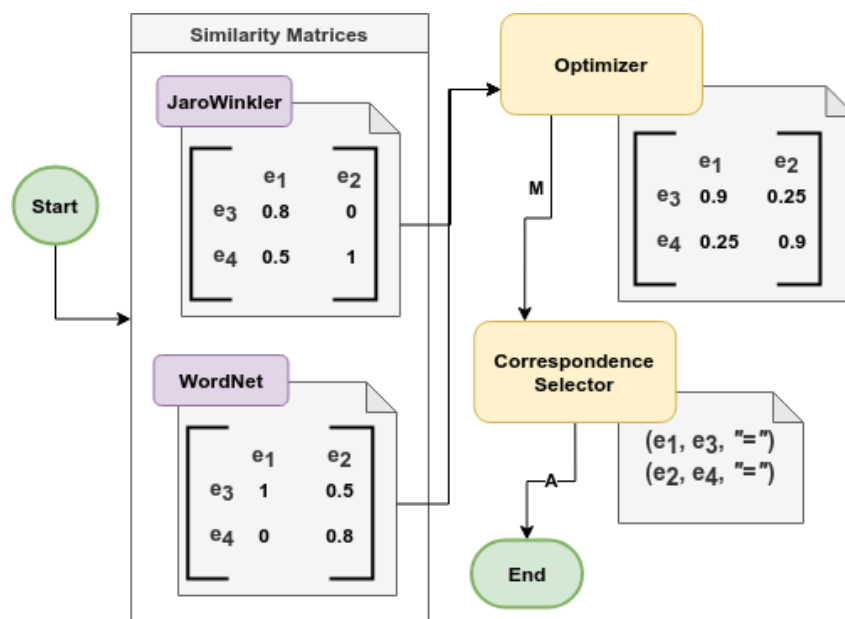
4.4.3 Correspondence Selection

The correspondence selection module is responsible for selecting the candidate matches of matrix M that are most likely to be a correct match. The algorithms used in this step vary according to the researcher's objective. The size of the ontologies also influences the choice of correspondence selection algorithms since the weights adjustment process of the optimizer is costly for large-scale ontologies. What is observed in the literature is that authors who work with large-scale ontologies allocate effort in the stage of correspondence selection, making use of

more complex algorithms such as memetics (XUE; LIU, 2017a), NSGA-III (XUE; LU; CHEN, 2019) and others (XUE et al., 2018a; XUE; CHEN, 2019c). While authors who work with conventional ontologies and focus on the optimization process use simpler selection algorithms such as greedy heuristics and other deterministic algorithms (SOUZA; SIQUEIRA; NUNES, 2019; ACAMPORA et al., 2013a). This framework allows the researcher to implement their own selection algorithms through a standard interface, whether they are specific heuristics or meta-heuristics. To propose a new solution, the researcher needs to implement a method that uses the M similarity matrix as an input and returns a set of matches that represents the final alignment.

To exemplify this process, Figure 22 shows the data flow starting from the confidence value calculated by each matcher for each candidate correspondence. If e_1 and e_2 are entities in the source ontology and e_3 and e_4 are entities in the target ontology, the experiment optimizer combines the confidence value while the match selector chooses the most reliable pairs. This framework provides three selection algorithms as examples. The first algorithm is an iterative greedy algorithm that seeks, at each iteration, the highest $\eta_{i,j}$ and selects the most reliable match. The second algorithm is based on the Hungarian method and details are presented in (KUHN, 1955). The third algorithm presents a Simulated Annealing metaheuristic implementation that, unlike the others, makes use of a structure to represent a solution based on correspondences. This structure is available for consumption in other algorithms and consists of a list of objects, each one containing the information about a single correspondence.

Figure 22 – Data Flow



Source: created by the author.

The algorithms available in the framework work with cardinality rates (1:1), which means, each entity can only be aligned once with another entity. Therefore, since two entities

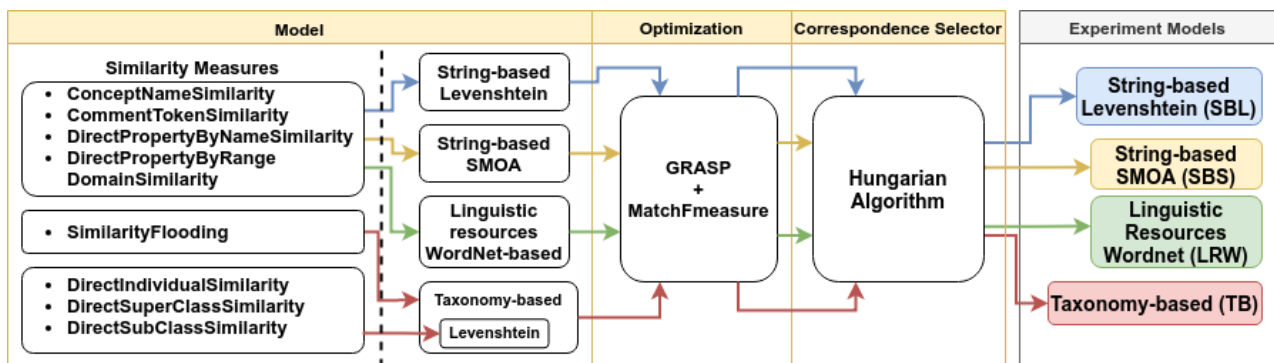
are chosen as a match, all other candidate matches that have one of these entities must be removed from matrix M . Other cardinalities can be added. The output format of this module is an alignment (A) that contains the most reliable correspondences in M .

4.5 EXPERIMENTAL PLANNING

The H1 hypothesis concerns the use of multiple similarity measures to verify the ability of finding correspondences for each type of measure. Groups were created based on the type of input that classifies the measure. Based on the classification model (EUZENAT; SHVAIKO, 2013), four groups were created that form the basis for four different experiment models presented in Figure 23. The first two models (SBL and SBS) represent terminological measures operating with two different similarity measures: Levenshtein distance (used in (XUE; WANG, 2017; ZHANG; GUO, 2019; FORSATI; SHAMSFARD, 2016)), and SMOA (XUE; PAN, 2017) which according to (STOILLOS; STAMOU; KOLLIAS, 2005) is the most performing measure for the OM problem. The third model (LRW) is also terminological, but adopts a linguistic approach based on WordNet. Finally, the last model (TB) includes measures at the structural level, represented mainly by the similarity flooding algorithm which is a Taxonomy-based approach.

The algorithms used in the Optimization and Correspondence Selection steps were chosen empirically in order to reduce stochastic factors that impact both the quality of the result and the time spent to obtain it. The main objective was to provide that the data collected from the experiment were as less dependent as possible on the Optimization and Correspondence Selection steps. It was used the Hungarian method (deterministic method), the MatchFmeasure (not influenced by reference alignments), and GRASP.

Figure 23 – Configurations created for evaluation of H1

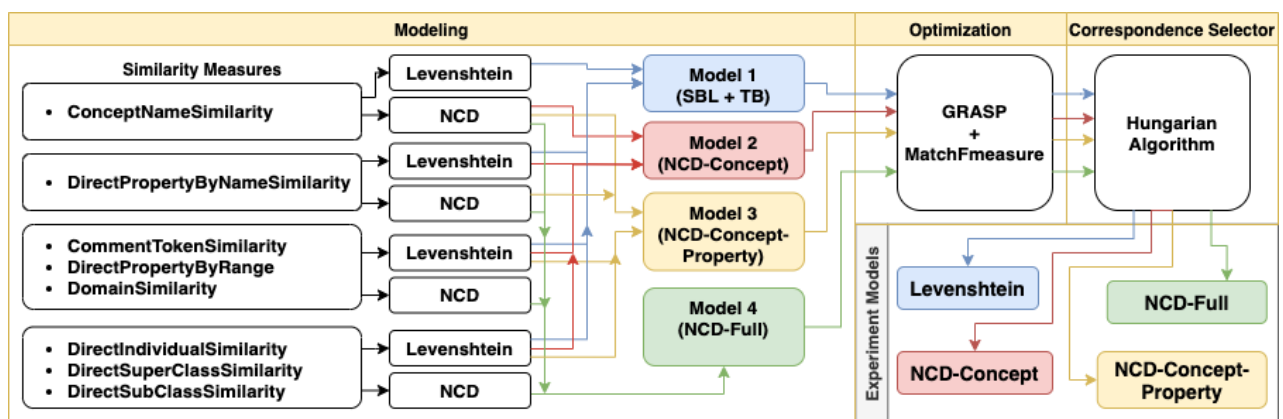


Source: created by the author.

Hypothesis H2 is also associated with the Modeling module. Four different experiment models were built: Levenshtein, NCD-Concept, NCD-Concept-Property, and NCD-Full (see Figure 24). The first one (Levenshtein) consists of the combination of similarity measures that were separated in H1, all using the Levenshtein distance (Model 1) as similarity calculation

strategy. The same set of measures was used in the second experiment model (NCD-Concept), however, with the NCD replacing the Levenshtein distance in the calculation of similarity between concept names (Model 2). In the NCD-Concept-Property model (Model 3), in addition to the concept name similarity, the NCD was also applied to property names. Finally, in the NCD-Full model (Model 4), all similarity measures are configured to use the NCD. The objective is to assess the impacts of the NCD in this scenario. As in H1, the Optimization and Correspondence Selection algorithms were chosen to minimize stochastic variations both in time and in the quality of the result so that the experiment actually portrays the behavior of similarity measures.

Figure 24 – Configurations created for evaluation of H2

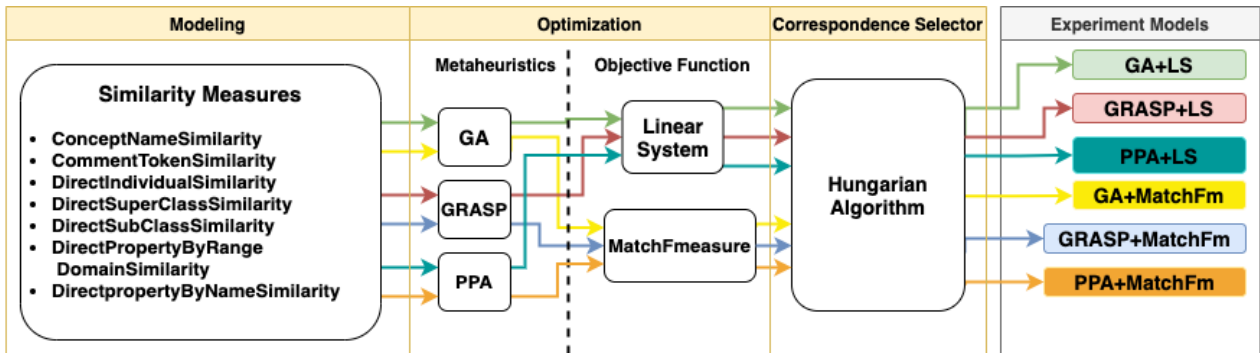


Source: created by the author.

The H3 hypothesis is associated with the use of different objective functions. To test it, two objective functions of the framework were selected, both of them can be used in the same application scenario, where a candidate alignment is sought, which will be validated later by a specialist. The objective functions are: maximizing the MatchFmeasure (unsupervised) and minimizing the error when solving the linear system (semi-supervised). Both objective functions were tested with three different meta-heuristics: GA, GRASP, and PPA. The details of each model can be seen in Figure 25. The Hungarian algorithm was used as a correspondence selector by the H3 experiment models in order to minimize the stochastic factors and allow the result to occur due to the variation of the objective and meta-heuristic functions in the optimization step. Also, the set of similarity measures is composed of measures that until then were separated in hypotheses H1 and H2 but which presented promising results when combined.

The H4 hypothesis suggests the comparison between different algorithms that can be executed in the correspondence selection step, with an emphasis on the Hungarian method (KUHN, 1955) and methods based on heuristics. To test it, two experiment models were created, both with the same algorithms in the Modeling and Optimization steps but varying the Correspondence Selection algorithm. In Figure 26, the Hungarian model uses the Hungarian method as a correspondence selector while the SA model uses an implementation of the Simulated

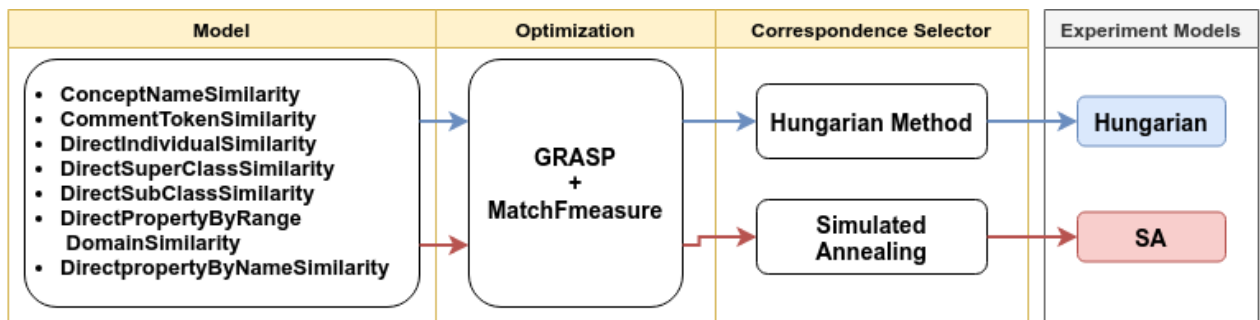
Figure 25 – Configurations created for evaluation of H3



Source: created by the author.

Annealing algorithm based on the proposal of (MOHAMMADI; HOFMAN; TAN, 2019).

Figure 26 – Configurations created for evaluation of H4



Source: created by the author.

4.6 EXPERIMENTAL RESULTS

Our experiments were carried out using one of the datasets provided by the Ontology Alignment Evaluation Initiative² (OAEI). Among the benchmarks made available by OAEI, the biblio benchmark is the one that better suits the ontology meta-matching problem. Each test case present in the biblio benchmark consists of the alignment between a source synthetic ontology and a target ontology. The source ontology is the same in each test case, while the target ontology varies according to the test number. In test ranges 1xx and 2xx, the target ontology is created systematically from the source ontology where, at each test, part of the ontology information is discarded/modified in order to evaluate how the algorithm behave when this information is lacking. In range 3xx, the alignment between the source ontology and real ontologies in the domain of bibliographic references is verified. This test base is suitable for meta-matching

² <http://oaei.ontologymatching.org/>

because what is expected from this type of approach is an adaptive behavior that rewards the most efficient solutions to maximize the quality of the result in different scenarios.

The dataset was divided into test subsets. The 1xx subset is simpler than the others, as it contains only generalization or language restriction misrepresentations. The 3xx subset contains all tests with real ontologies. Tests 2xx, which contains misrepresentations related to hierarchy, instances, nomenclature, classes and others, was subdivided into two subsets. The subset 2xx-1 contains all tests that have one or two misrepresentations, while subset 2xx-2 contains test cases that have more than two misrepresentations. All tests were conducted by an Ubuntu 14.04.5 LTS machine with Intel Xeon CPU E5-2650 v2 @ 2.60GHz and 16GB RAM.

4.6.1 H1 Evaluation

As mentioned before, H1 hypothesis concerns the use of multiple similarity measures to verify the ability of finding correspondences for each type of measure. Table 20 shows the accuracy by subset of each model built for the H1 test, in the 1xx range all models achieved the maximum result, while in the others the best results were with the SBL and TB models. Based on this preliminary result, a model composed of measures from the SBL and TB models was created, the results of this model are also presented in Table 20. The results achieved by the SBL + TB model equal or exceed the others in terms of F-measure. In some test cases, such as T258 and T259 in Figure 27, neither SBL nor TB achieve the best result, but the combination of both allowed new correspondences to be found. Tables 21 and 22 show the Jaccard coefficient between the SBL, TB, and SBL + TB models for the T209 and T258 tests, respectively. The higher the Jaccard coefficient, the more correspondences the experiments found in common, whether they were correct or not. In the T209 test, SBL achieved the best results followed by TB, the SBL + TB model was able to surpass the others in T209 and, according to the Jaccard coefficient, the matches found by SBL + TB are more similar to those of TB, which indicates a greater contribution by TB measures even SBL having been higher than TB in this case. In the T258 test, where neither SBL nor TB did well but SBL + TB outperformed the others, the Jaccard coefficient shows that new matches were found, since the correlation between the experiments is low.

In terms of execution time, the composite model of SBL + TB was also more expensive than the others, since it has a larger set of similarity measures. Figure 28 shows the mean execution time in seconds of each model for each test case, the LRW model ends up spending more time compared to the other terminological models because it requires an external connection to search on WordNet. The model composed by SBL + TB spent an average of 37 seconds, while the SBL model and the TB model spent an average of 21 and 17 seconds, respectively.

To compare the results achieved by the composite model of SBL + TB with results from the literature, two studies were selected whose evaluation uses the same OAEI benchmark and the results are divided by the test number. Table 23 shows the results made available by

Table 20 – F-measure rates for the four experiment models

	#Test	Precision	Recall	F-measure
String-based Levenshtein (SBL)	1xx	1.000	1.000	1.000
	2xx-1	0.923	0.947	0.931
	2xx-2	0.238	0.268	0.248
	3xx	0.808	0.807	0.804
String-based SMOA (SBS)	1xx	1.000	1.000	1.000
	2xx-1	0.897	0.921	0.905
	2xx-2	0.236	0.268	0.247
	3xx	0.755	0.757	0.754
Linguistic Resources Wordnet (LRW)	1xx	1.000	1.000	1.000
	2xx-1	0.894	0.918	0.903
	2xx-2	0.245	0.276	0.255
	3xx	0.652	0.660	0.653
Taxonomy-based (TB)	1xx	1.000	1.000	1.000
	2xx-1	0.869	0.893	0.877
	2xx-2	0.336	0.379	0.351
	3xx	0.595	0.596	0.594
Composition of SBL + TB	1xx	1.000	1.000	1.000
	2xx-1	0.944	0.967	0.952
	2xx-2	0.407	0.452	0.423
	3xx	0.808	0.807	0.804

Source: created by the author.

Table 21 – Jaccard Coefficient for H1 test number 209

Jaccard #209	SBL	TB
SBL + TB	0.500	0.803
SBL	1.000	0.422

Source: created by the author.

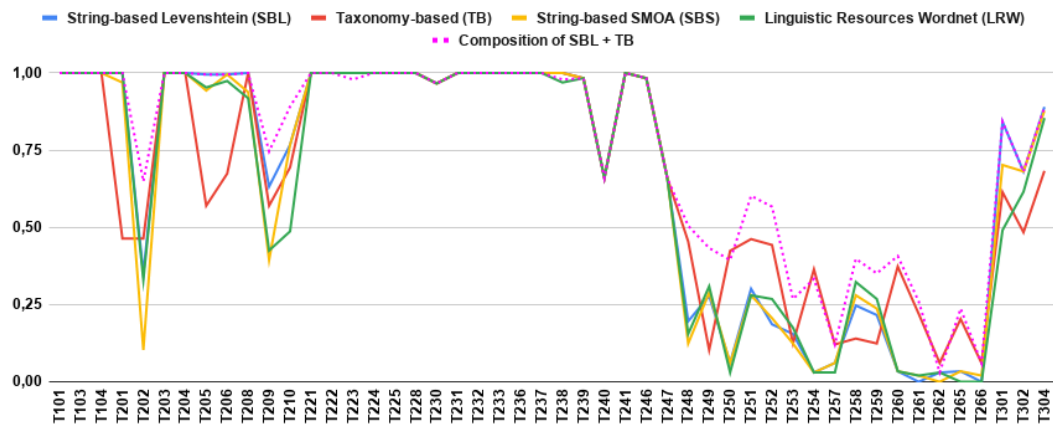
Table 22 – Jaccard Coefficient for H1 test number 258

Jaccard #258	SBL	TB
SBL + TB	0.155	0.069
SBL	1.000	0.075

Source: created by the author.

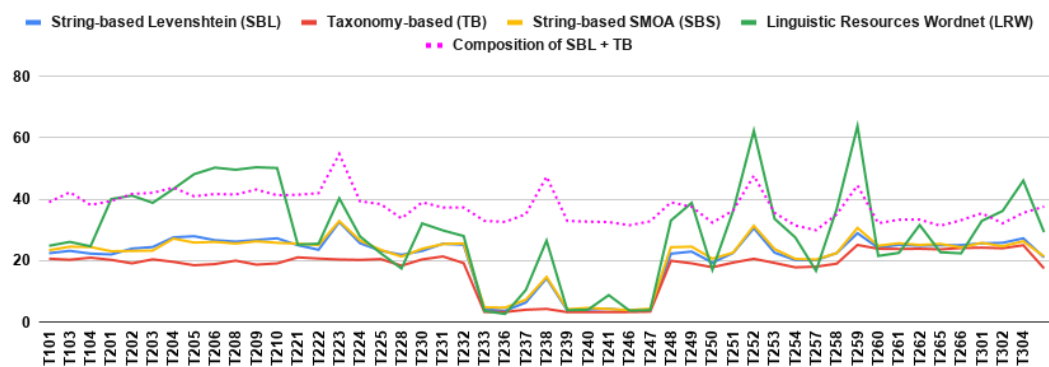
articles (XUE et al., 2015) and (XUE; WANG, 2017), as well as the result achieved by the SBL + TB model for the same test cases. In this case, the H1 hypothesis proves to be true, because when using String-based, Taxonomy-based, and Linguistic Resources measures in an isolated environment, which does not suffer variations in the algorithms of other processing steps, it was possible to achieve results compatible with those in the literature. The studies pointed to different groups of correspondences associated with each set of similarity measures that, when

Figure 27 – Best F-measure rate for each H1 model, for each test case



Source: created by the author.

Figure 28 – Mean runtime in seconds of each model for each test case



Source: created by the author.

combined, allow the creation of a more resilient tool, albeit slower.

4.6.2 H2 Evaluation

Hypothesis H2 tests the use of NCD as a new similarity measure applied in OMM. Table 24 shows the accuracy results of the models that evaluate the NCD, it is also possible to observe in how many tests, of each specific range, a model achieved the best result among all models (last column). The results of the NCD varied, in the 3xx test range that includes alignments with real ontologies, there was no result capable of surpassing the Levenshtein model (SBL + TB), however in the 2xx-1 and 2xx-2 ranges there were some test cases whose result was higher, mainly in the 2xx-2 range of the NCD-Concept-Property model, where eight test cases were better than the others. Despite the improvement in some specific test cases, the average value points to compensation in the other cases causing the difference to appear in the third decimal place. Figure 29 shows that the use of NCD in some of the similarity measures did

Table 23 – Comparison of SBL + TB with literature approaches in terms of F-measure

Test	(XUE et al., 2015)	(XUE; WANG, 2017)	Composition of SBL+TB
101	1,000	1,000	1,000
103	1,000	1,000	1,000
104	1,000	1,000	1,000
201	0,940	0,940	1,000
203	0,990	0,990	1,000
204	0,980	0,980	1,000
205	0,930	0,930	0,995
206	0,700	0,700	0,995
221	1,000	1,000	1,000
222	1,000	1,000	1,000
223	0,990	0,990	0,979
224	1,000	1,000	1,000
225	1,000	1,000	1,000
228	1,000	1,000	1,000
230	1,000	1,000	0,966
231	1,000	1,000	1,000
301	0,750	0,810	0,842
302	0,740	0,850	0,681
304	0,930	0,930	0,890
Mean	0,945	0,954	0,966

Source: created by the author.

not impact on large differences in the experiment execution time, in the first half of the test cases the intermediate models obtained better performance than Levenshtein, a situation that is reversed from the test T258 ahead. In general, the NCD demonstrated that it can be a metric to be explored in OMM approaches, the H2 hypothesis was found to be true in some of the test cases but not in the vast majority.

4.6.3 H3 Evaluation

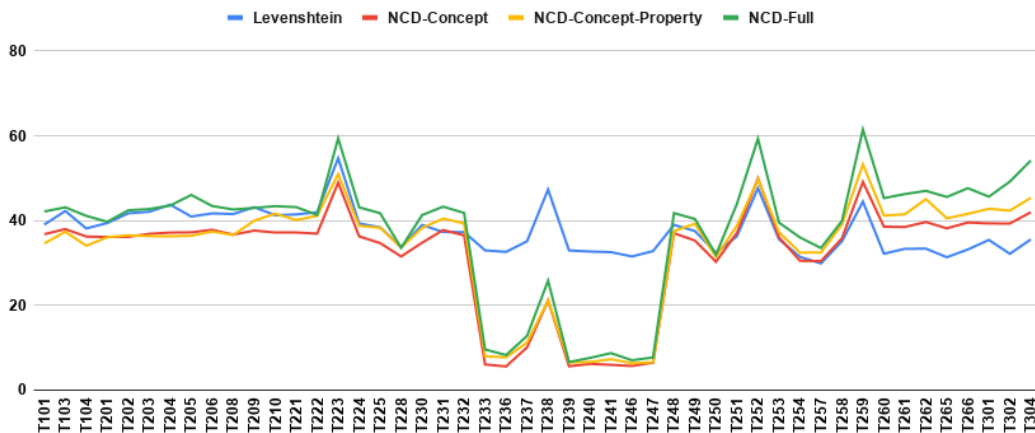
In hypothesis H3, three meta-heuristics and two distinct objective functions were evaluated, totaling six experiment models. Table 25 shows the accuracy result of each model, of the models that used the function based on the linear system (LS), GRASP presented better results in the ranges 2xx-1 and 3xx while GA and PPA were better in the range 2xx-2. The replacement of the LS function by MatchFm resulted in a jump in the quality of the result, mainly for the GRASP and PPA models, demonstrating that this objective function alone has the potential to show new points in the solution space. Among the models that used MatchFmeasure, GRASP and PPA obtained, on average, better results than GA, where GRASP was better in the 3xx range and the PPA was better in the 2xx-2 and 2xx-1 ranges. However, Figure 30 shows that PPA + MatchFm was much more expensive than GRASP + MatchFm in the vast majority of test cases, spending more than twice the time, which shows that in the experiments conducted

Table 24 – F-measure rates for the four experiment models of H2

	#Test	Precision	Recall	F-measure	Cases with the best result
Levenshtein	1xx	1.000	1.000	1.000	0
	2xx-1	0.944	0.967	0.952	2
	2xx-2	0.407	0.452	0.423	5
	3xx	0.808	0.807	0.804	3
NCD-Concept	1xx	1.000	1.000	1.000	0
	2xx-1	0.943	0.967	0.951	3
	2xx-2	0.411	0.457	0.426	7
	3xx	0.800	0.798	0.796	2
NCD-Concept-Property	1xx	1.000	1.000	1.000	0
	2xx-1	0.933	0.957	0.941	1
	2xx-2	0.413	0.461	0.429	8
	3xx	0.753	0.756	0.752	0
NCD-Full	1xx	1.000	1.000	1.000	0
	2xx-1	0.937	0.961	0.954	2
	2xx-2	0.402	0.448	0.417	2
	3xx	0.746	0.749	0.744	1

Source: created by the author.

Figure 29 – Mean runtime (seconds) of each H2 model for each test case



Source: created by the author.

GRASP had the best cost/benefit between the approaches that used MatchFm. Despite showing improvement in the quality of the results, the experiments that used MatchFm were more costly in terms of execution time than the LS, spending more than twice the time. Therefore, if the execution time is a determining factor in the use of the tool, it may be worthwhile to select some input correspondences for the construction of the linear system. Although they can be obtained more quickly, the results of the models with LS vary more as pointed out by Figure 31 and Figure 32. Figures 31 and 32 show the mean and standard deviation for each test case, indicating that in most tests at switching from LS to MatchFm the stability of the experiment increased.

Stability can be associated with the set of correspondences that are analyzed by the objective function, since MatchFm takes into account all possible pairs of entities while the LS models use a reference set that has 3 to 4% of the total correct correspondences.

The experiments show that each algorithm has its pros and cons, both an objective function and a meta-heuristic are capable of providing leaps in terms of quality and execution time, leaving the researcher to adapt them to their usage scenario. The H3 hypothesis was proven true for the tested algorithms because the quality of the solutions found as a result of MatchFm was close to the solutions found with LS models, surpassing the LS-based models in numerous test cases. In fact, it takes more time to find these solutions since there is no prior knowledge embedded in the objective function that limits the evaluation scope, which means that the unsupervised approach checks for more possibilities and spends more time.

Table 25 – F-measure rates for the six experiment models of H3

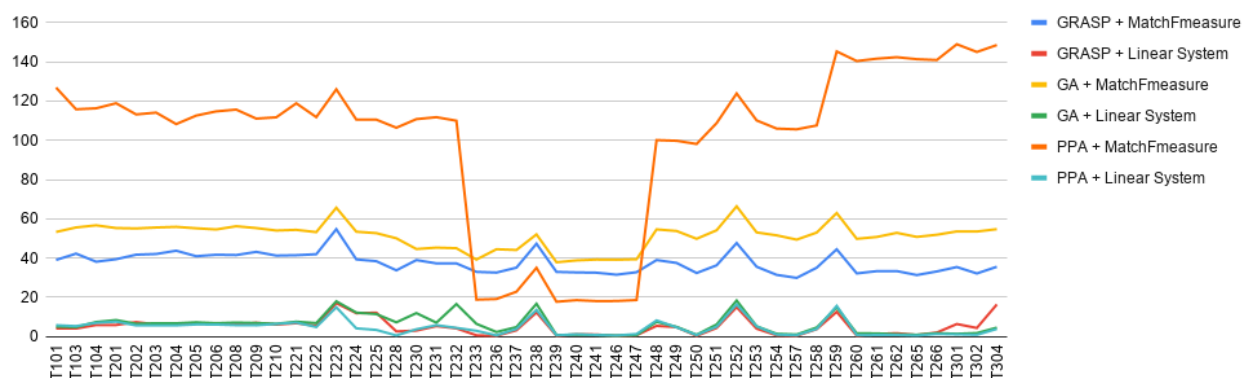
	#Test	Precision	Recall	F-measure
GA+LS	1xx	1.000	1.000	1.000
	2xx-1	0.934	0.957	0.942
	2xx-2	0.402	0.443	0.416
	3xx	0.604	0.613	0.607
GRASP+LS	1xx	1.000	1.000	1.000
	2xx-1	0.940	0.964	0.948
	2xx-2	0.358	0.398	0.372
	3xx	0.799	0.800	0.797
PPA+LS	1xx	1.000	1.000	1.000
	2xx-1	0.925	0.948	0.933
	2xx-2	0.399	0.443	0.414
	3xx	0.547	0.556	0.550
GA+MatchFm	1xx	1.000	1.000	1.000
	2xx-1	0.939	0.963	0.948
	2xx-2	0.353	0.395	0.367
	3xx	0.796	0.796	0.793
GRASP+MatchFm	1xx	1.000	1.000	1.000
	2xx-1	0.944	0.967	0.952
	2xx-2	0.407	0.452	0.423
	3xx	0.808	0.807	0.804
PPA+MatchFm	1xx	1.000	1.000	1.000
	2xx-1	0.945	0.969	0.954
	2xx-2	0.416	0.462	0.432
	3xx	0.794	0.794	0.791

Source: created by the author.

4.6.4 H4 Evaluation

In H4, two algorithms for correspondences selection were evaluated: the Hungarian method and SA metaheuristic. Table 26 presents the Precision, Recall, and F-measure averages

Figure 30 – Mean runtime (seconds) of each H3 model for each test case

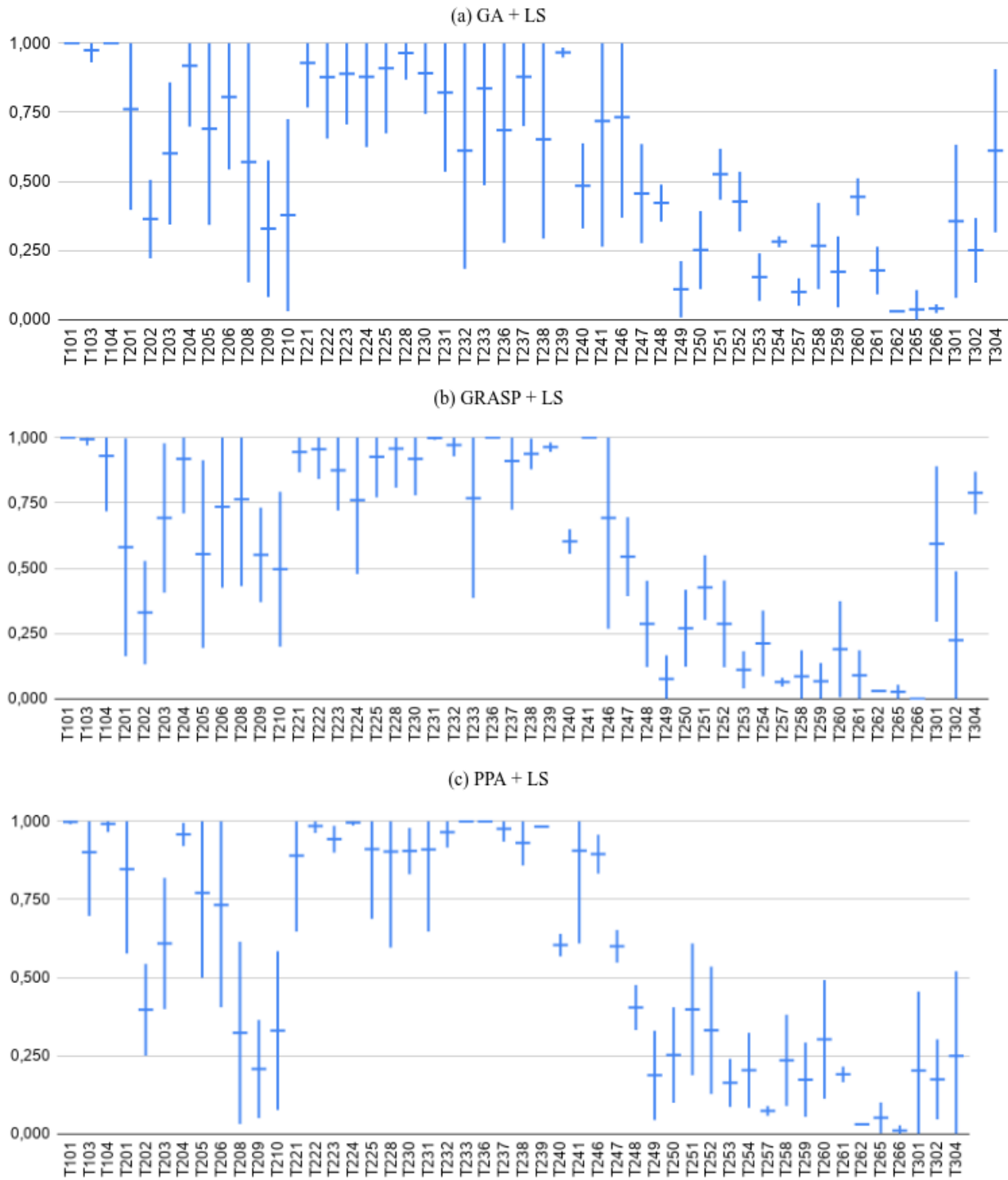


Source: created by the author.

for each test range. The two models showed similar results, the Hungarian method was slightly better in the 2xx-1 and 2xx-2 ranges while the SA was better in the 3xx range. Table 26 also counts the number of tests for each range where one model was better than the other, demonstrating that there was a lot of alternation in the dominance of the 2xx-2 range and that although the SA obtained better results in more tests in this range, the difference in cases where the Hungarian method dominated was greater since the mean of the SA was lower. The results of Table 26 show that the stochastic factors related to heuristics allow new correspondences to be found in the same way that it is possible to lose other correspondences.

Figure 33 shows the mean execution time spent by each model in each test case, in the total mean, SA spent approximately 35 seconds while Hungarian spent approximately 38 seconds per test. The data in Figure 33 points to similar results in the first test cases and greater variations in the second and third half. When analyzed together, the data in Table 26 and Figure 33 show that algorithms with approximate solutions such as SA have the potential to achieve results close to optimal algorithms such as Hungarian. Figure 33 pointed to similar times in several test cases, but heuristic algorithms like SA have parameters that can be configured to reduce processing time, which can be useful in matching large-scale ontologies, where the number of entities is significantly higher than those in the OAEI biblio benchmark. The heuristic algorithms also have the advantage of being able to work with different objective functions that can embed other parameters to be optimized, while the Hungarian method works in a more restricted way using the final confidence value that is assigned to each candidate correspondence. The H4 hypothesis proved to be true, since there were test cases where the SA obtained better results than the Hungarian and in a similar time, which can still be improved through the optimization of the SA configuration parameters.

Figure 31 – Mean F-measure and deviation of H3 linear system based experiment models

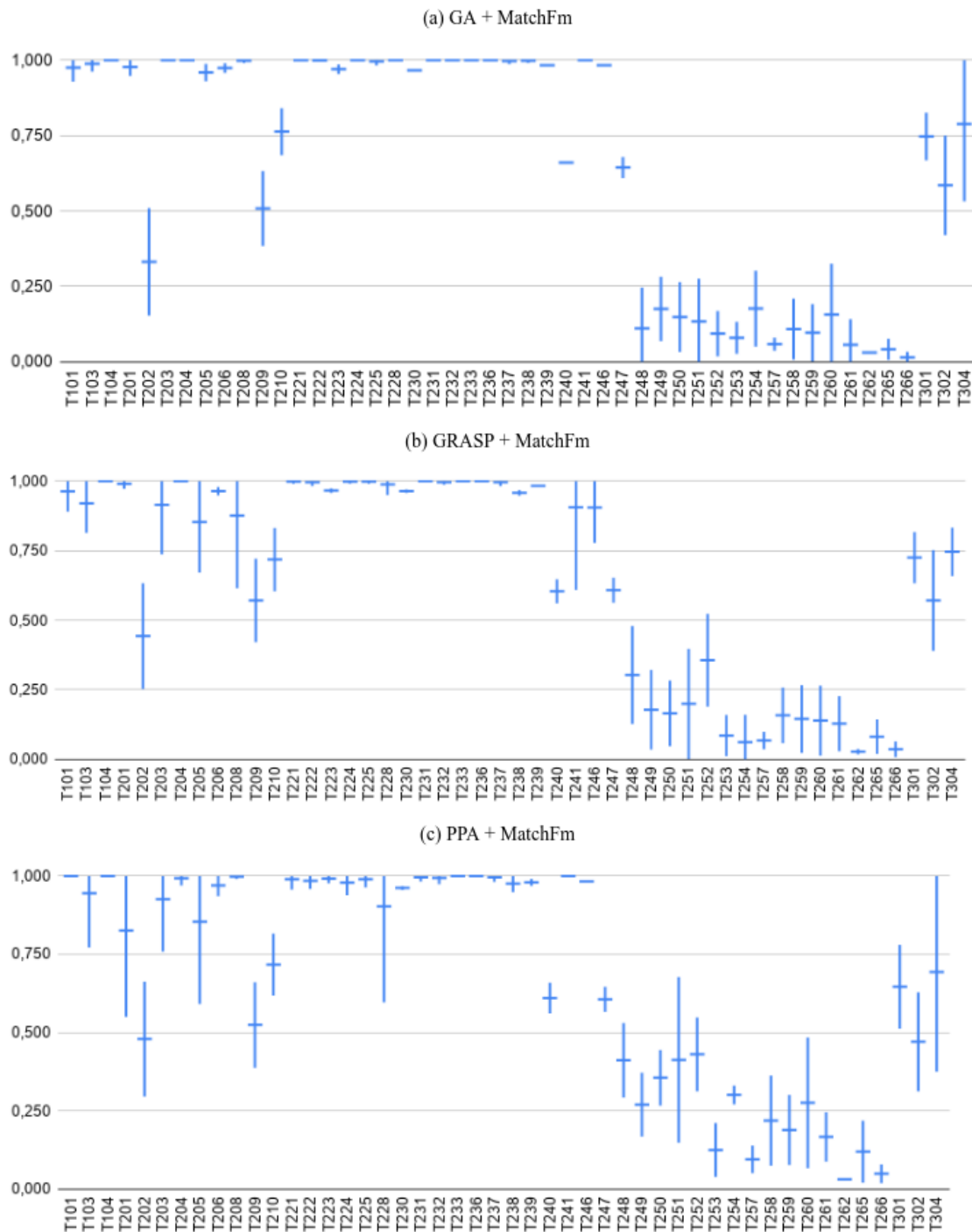


Source: created by the author.

4.7 CONCLUDING REMARKS

This manuscript presented an analysis of algorithms commonly used in the processing steps of OMM approaches. The main objective was to evaluate the cost / benefit of the algorithms

Figure 32 – Mean F-measure and deviation of MatchFm based experiment models



Source: created by the author.

tested in a controlled environment, that is, to compare experiments with small modifications in the same processing step. This methodology allowed to identify the contribution of each algorithm used in different parts of a compound solution to the problem of OMM.

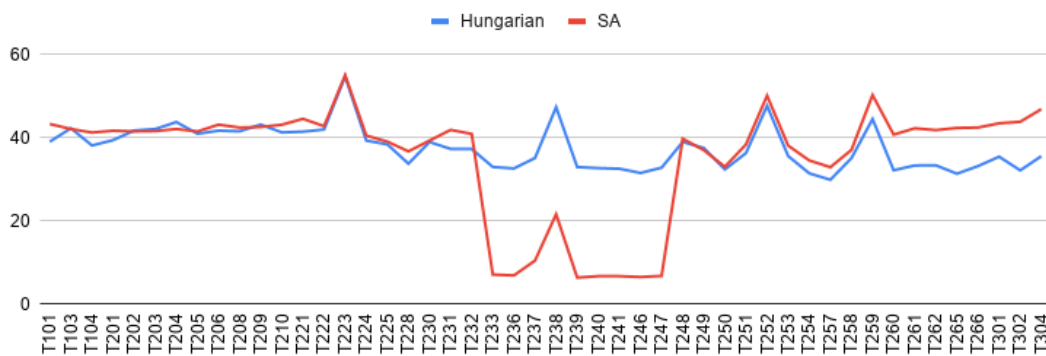
The results can be divided by the three processing steps: Modeling, Optimization, and Correspondence Selection. In the Modeling stage, it could be verified that the more characteristics

Table 26 – F-measure rates for the two experiment models of H4

	#Test	Precision	Recall	F-measure	Cases with the best result
Hungarian	1xx	1.000	1.000	1.000	0
	2xx-1	0.944	0.967	0.952	3
	2xx-2	0.407	0.452	0.423	6
	3xx	0.808	0.807	0.804	0
SA	1xx	1.000	1.000	1.000	0
	2xx-1	0.939	0.963	0.947	2
	2xx-2	0.399	0.444	0.414	7
	3xx	0.812	0.811	0.808	1

Source: created by the author.

Figure 33 – Mean runtime (seconds) of each H4 model for each test case



Source: created by the author.

of an entity can be captured by a similarity measure, the better the quality of the tool, with emphasis on the results achieved by taxonomic similarity measures and string-based measures using the Levenshtein distance. The results showed that even if two sets of similarity measures do not achieve good results when used individually, it does not mean that they should be discarded, since their combination can provide new correspondences. Still on the Modeling stage, experiments with the NCD reaffirmed that there is no technique that stands out from the others in all cases, since there was improvement in some cases and worsening in others, which makes the option of using the metric be left to the researcher based on the characteristics of the ontologies in alignment.

In the optimization stage, the results showed that algorithms with a focus on local optimization have the potential to find good solutions to the problem, as was the case with GRASP that surpassed the others in several cases and with different objective functions. Of the objective functions studied, MatchFmeasure showed results with higher quality and stability in multiple executions, however the function based on the linear system is faster, spending approximately less than half the time of MatchFmeasure. In the correspondence selection stage, it was seen that both deterministic and heuristic methods can find good solutions to the

problem, while deterministic ones present stable solutions, heuristics have variations that can save processing time and find different groups of correspondences.

A tool for experimenting with OMM approaches was built to conduct this study and is available for future research³, this tool can be configured with new algorithms to test other research questions. The experiments conducted in this work used only mono-objective objective functions, however the tool architecture supports and facilitates the creation of multi-objective approaches, opening a new range of options that allows the construction of more robust algorithms with greater ability to visualize the solution space.

The results of this study were obtained using the OAEI biblio benchmark. OAEI has several other evaluation bases to test OM approaches and some characteristics of these bases can influence the result of the studied algorithms. One of the datasets available at OAEI concerns the alignment of large scale ontologies, which increases the difficulty due to the large volume of data, this factor may make the use of some of the methods studied in this research unfeasible. In future works, it is intended to extend the analysis to other bases, verifying the variation of results.

³ <https://bitbucket.org/nicolasferranti/heuristicontologymatching/src/master/>

5 CONCLUSIONS

This master thesis presented a study on ontology meta-matching approaches, focusing on the use of meta-heuristics. In the first manuscript (Chapter 2), a systematic mapping of the literature was performed in order to identify how the problem is modeled, and which are the main algorithms that researchers use to address it. It was shown that the OMM workflow can be divided into three main steps (Modeling, Optimization and Correspondence Selection), and there are several algorithms that can be applied in each step of the workflow, with emphasis on the use of meta-heuristics in some of the stages. The data from the systematic mapping also pointed out that the metrics Precision, Recall, and F-measure are the most common to evaluate the experiments. However, when an author compares his proposal with another, it is not clear what impact each algorithm had on the construction of the solution, since the algorithms used in each of the three steps are completely different.

Based on this premise, we presented a framework in Chapter 3 that was designed to simplify the scientific experimentation process in OMM. The framework is divided into modules and allows different algorithms to be tested at each stage of the workflow without compromising the rest of the execution of the experiment, which facilitates the development and testing of independent algorithms, associated with a single step of the workflow. Some of the main and most common algorithms founded in the systematic mapping are available in the framework. Researchers can make use of existing algorithms or propose their own approaches without major difficulties, since the framework is an open source tool.

Finally, a series of experiments was conducted in order to better understand the differences between the algorithms that are used in the same step of the workflow. The experiments were conducted following the premise of maintaining the same algorithms in the other stages of the workflow and applying specific variations in a specific step. The framework is capable of generating various information about the experiment through a provenance module, that works automatically. In Chapter 4, the results are compared according to the accuracy, experiment runtime, objective functions runtime, rate of correlation between experiments, solution stability, and other information that helps to understand the behavior of a technique.

Through this methodology it was possible to conclude that, among the most common groups of similarity measures used in the literature (String-based, Taxonomy-based, and Linguistic Resources), a combination of String-based and Taxonomy-based measures was enough to create a model with good adaptive capacity that found results close to other approaches in the literature in the same dataset. Still on similarity measures, tests conducted with a similarity measure hitherto not used in OMM reaffirmed that there is no similarity measure that stands out from the others in all cases. Although the genetic algorithm is the most used in the literature, there is space for using local search algorithms such as GRASP, which, depending on the context of use, can present better solutions in less time. The comparison between different objective

functions highlights the relevance that these algorithms have both in the execution time of an experiment and in the quality of a result, where methods that take into account all entities of ontologies tend to be more costly and to build more stable solutions, while methods using a subset of entities are computationally faster but produce more unstable solutions.

5.1 LIMITATIONS

The work limitations can be divided into three parts: (i) mapping conduction, (ii) usability of the framework, and (iii) experimental results.

Mapping conduction. The systematic literature mapping aimed to identify challenges in ontology meta-matching domain. However, as any method, there are threats to its validity and limitations. The results of this study may have been influenced by certain uncontrollable limitations. Although the mapping covered 51 articles, removing articles not written in English and those in gray literature, for example, diminished the accuracy of the conclusions. During the selection process some influential work in the area may have been lost, however this mapping focused on articles whose explanation of the main components of the OMM process was clearly discussed. Furthermore, not all relevant electronic databases were considered, i.e. ACM Digital Library, so it is possible that relevant studies were not indexed in the selection of this mapping. However, the mapping relies on the representativeness of the repositories selected to answer the research questions.

Framework usability. The framework was developed using some API in Java to manipulate ontologies and RDF data, such as Apache Jena and OWL API. Therefore, the framework can be used with OMM approaches coded in Java. The framework defines an execution workflow that does not allow changes to the execution flow. However, this execution flow proved to be comprehensive enough to cover the types of approaches found in systematic mapping.

Experimental Results. The experiments were conducted with a single OAEI dataset, so there are still several other questions to be answered, such as, for example, the application of these algorithms for the alignment of large-scale ontologies, which may result in the impossibility of using some algorithms. Factors such as the execution time become fundamental in this type of scenario because there are tens of thousands of classes to be analyzed; certainly a more in-depth study on this issue is necessary. Although the experiments were based on common algorithms in the literature, there are still several algorithms that can be applied and evaluated in this scenario. Some of the algorithms that have been tested are stochastic and have a set of parameters that must be configured to optimize usage. In this work, the algorithms were configured empirically and following suggestions from other authors, there are other techniques for adjusting parameters that can be used and possibly build better configurations. This work sought to analyze the impact of each algorithm in terms of runtime and output alignment. Analyses focusing on specific characteristics of an algorithm, such as the convergence rate, can help to better understand others aspects of its behaviour.

5.2 CONTRIBUTIONS

The main contributions of this work are:

- A systematic mapping on metaheuristics-based OMM approaches covering seven years of published researches. The study presents a map of the implementation details usually used in unsupervised and supervised OMM approaches. To the best of our knowledge, it is the first systematic mapping on OMM approaches. Among the contributions of the mapping are: (i) the identification of relevant articles for the area highlighted in the citations graph, (ii) the identification of the most common types of similarity measures among the articles, (iii) the recognition of the most common meta-heuristics used to address the problem, as well as detailing how solutions are represented, (iv) the study of the evolution of the types of objective function over the years, and (v) the identification of the OAEI datasets used by the authors and which result achieved by each work.
- A modularized framework designed for scientific experimentation in OMM that already has the main algorithms used by researchers and that facilitates the process of implementing new approaches. The implementation is freely available for further research¹. The creation of a data provenance model that provides explanations of the results of the experiment contributing to increased reliability and allowing new analyzes to be performed from different perspectives.
- A novel PPA-based approach for OMM. The implementation is available in the framework and its results were detailed in (FERRANTI; SOARES; SOUZA, 2018b).
- A comparative case study between the main solutions applied in dealing with the problem of OMM; Results showed that, the more characteristics of the entities that can be captured by similarity measures set, the greater the accuracy of the model. It was also possible to observe the good performance and accuracy of local search based meta-heuristics when compared to global optimization meta-heuristics. Experiments with different objective functions have shown that semi-supervised methods can shorten the execution time of the experiment but, on the other hand, bring more instability to the result.

This work opens a range of possibilities for future research. The architecture of the framework allows new algorithms to be coupled and tested. One aspect with potential results is the construction of experimental models with multi-objective functions. The framework already has eight distinct objective functions that can be combined and parameterized by a meta-heuristic. The main challenge is to work with multi-objective meta-heuristics because in this scenario there is not only a single optimal solution, but a set of non-dominated solutions.

¹ <<https://bitbucket.org/nicolasferranti/heuristicontologymatching/src/master/>>

As mentioned earlier, conducting experiments on other OAEI datasets can lead to more general conclusions about the studied algorithms as well as leading to new challenges that will allow the evolution the framework to an increasingly comprehensive tool. Although the OAEI biblio benchmark is the most suitable dataset for OMM, the systematic mapping showed that some authors have used other datasets with promising results, which can serve as a reference for the replication of this work using these ontologies.

APPENDIX A - LIST OF PAPERS AND IDENTIFICATORS

Table 27 – List of papers used in Figure 6

ID	Ref	ID	Ref
1	(ACAMPORA; LOIA; VITIELLO, 2013)	2	(ACAMPORA et al., 2013a)
3	(BINIZ; AYACHI, 2018)	9	(KUREYCHIK; SEMENOVA, 2017)
10	(MARTÍNEZ-ROMERO et al., 2013)	11	(MOHAMMADI; HOFMAN; TAN, 2018)
12	(RAMESH et al., 2016)	13	(SEMENOVA; KUREYCHIK, 2016a)
14	(SEMENOVA; KUREYCHIK, 2016b)	15	(XUE; WANG; HAO, 2014)
16	(XUE; WANG; REN, 2014)	17	(XUE et al., 2015)
18	(XUE; WANG; HAO, 2015)	19	(XUE; WANG, 2015a)
20	(XUE; WANG, 2015b)	21	(XUE et al., 2015)
22	(XUE; LIU, 2017c)	23	(XUE; WANG, 2017)
24	(XUE; WANG; REN, 2016)	25	(XUE; TSAI; FENG, 2017)
26	(XUE, 2017)	27	(XUE; TSAI; ZHANG, 2016)
28	(XUE; PAN, 2017)	29	(XUE; TSAI; WANG, 2017)
30	(XUE; TSAI; FENG, 2017)	31	(XUE; LIU, 2017a)
32	(XUE; LIU, 2017b)	33	(XUE et al., 2018b)
34	(XUE; CHEN, 2018)	35	(XUE; REN, 2017)
36	(XUE; LIU; WANG, 2017)	37	(XUE; PAN, 2018)
38	(XUE; HANG; TANG, 2019)	39	(XUE; LIU, 2018)
40	(XUE; CHEN; CHEN, 2019)	41	(XUE; CHEN, 2019b)
42	(XUE; CHEN, 2019c)	43	(XUE et al., 2018a)
44	(XUE et al., 2018c)	45	(XUE; CHEN, 2019a)
46	(XUE; LU; CHEN, 2019)	47	(ZHANG; GUO, 2019)
48	(FERRANTI; SOARES; SOUZA, 2018a)	49	(XUE; CHEN; YAO, 2018)

REFERENCES

- ACAMPORA, Giovanni; ISHIBUCHI, Hisao; VITIELLO, Autilia. A comparison of multi-objective evolutionary algorithms for the ontology meta-matching problem. In: IEEE. **2014 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2014. p. 413–420.
- ACAMPORA, Giovanni; KAYMAK, Uzay; LOIA, Vincenzo; VITIELLO, Autilia. Applying nsga-ii for solving the ontology alignment problem. In: IEEE. **Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on**. [S.l.], 2013. p. 1098–1103.
- ACAMPORA, Giovanni; KAYMAK, Uzay; LOIA, Vincenzo; VITIELLO, Autilia. A fml-based fuzzy tuning for a memetic ontology alignment system. In: IEEE. **2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. [S.l.], 2013. p. 1–8.
- ACAMPORA, Giovanni; LOIA, Vincenzo; VITIELLO, Autilia. Enhancing ontology alignment through a memetic aggregation of similarity measures. **Information Sciences**, Elsevier, v. 250, p. 1–20, 2013.
- ACAMPORA, Giovanni; VITIELLO, Autilia. A study on local search meta-heuristics for ontology alignment. In: **Computational Intelligence for Semantic Knowledge Management**. [S.l.]: Springer, 2020. p. 53–70.
- ARDJANI, Fatima; BOUCHIHA, Djelloul; MALKI, Mimoun. Ontology-alignment techniques: survey and analysis. **International Journal of Modern Education and Computer Science**, Modern Education and Computer Science Press, v. 7, n. 11, p. 67, 2015.
- BANDROWSKI, Anita; BRINKMAN, Ryan; BROCHHAUSEN, Mathias; BRUSH, Matthew H; BUG, Bill; CHIBUCOS, Marcus C; CLANCY, Kevin; COURTOT, Mélanie; DEROM, Dirk; DUMONTIER, Michel et al. The ontology for biomedical investigations. **PloS one**, Public Library of Science, v. 11, n. 4, 2016.
- BANERJEE, Satanjeev; PEDERSEN, Ted. Extended gloss overlaps as a measure of semantic relatedness. In: **Ijcai**. [S.l.: s.n.], 2003. v. 3, p. 805–810.
- BERNERS-LEE, Tim; HENDLER, James; LASSILA, Ora et al. The semantic web. **Scientific american**, New York, NY, USA:, v. 284, n. 5, p. 28–37, 2001.
- BINIZ, Mohamed; AYACHI, Rachid El. Optimizing ontology alignments by using neural nsga-ii. **Journal of Electronic Commerce in Organizations (JECO)**, IGI Global, v. 16, n. 1, p. 29–42, 2018.
- BOCK, Jürgen; HETTENHAUSEN, Jan. Discrete particle swarm optimisation for ontology alignment. **Information Sciences**, Elsevier, v. 192, p. 152–173, 2012.
- BUDANITSKY, Alexander; HIRST, Graeme. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In: **Workshop on WordNet and other lexical resources**. [S.l.: s.n.], 2001. v. 2, p. 2–2.
- CHONDROGIANNIS, E.; ANDRONIKOU, Vassiliki; KARANASTASIS, E.; VARVARIGOU, Theodora. An intelligent ontology alignment tool dealing with complicated mismatches. **CEUR Workshop Proceedings**, v. 1320, 01 2014.

- DAMERAU, Fred J. A technique for computer detection and correction of spelling errors. **Communications of the ACM**, ACM New York, NY, USA, v. 7, n. 3, p. 171–176, 1964.
- DJEDDI, Warith Eddine; KHADIR, Mohamed Tarek. Ontology alignment using artificial neural network for large-scale ontologies. **International Journal of Metadata, Semantics and Ontologies 16**, Inderscience Publishers Ltd, v. 8, n. 1, p. 75–92, 2013.
- ESSAYEH, Aroua; ABED, Mourad. Towards ontology matching based system through terminological, structural and semantic level. **Procedia computer science**, Elsevier, v. 60, p. 403–412, 2015.
- EUSUFF, Muzaffar; LANSEY, Kevin; PASHA, Fayzul. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. **Engineering optimization**, Taylor & Francis, v. 38, n. 2, p. 129–154, 2006.
- EUZENAT, Jérôme; SHVAIKO, Pavel. **Ontology matching**. 2nd edition. ed. [S.l.]: Springer, 2013.
- EUZENAT, Jérôme; SHVAIKO, Pavel et al. **Ontology matching**. [S.l.]: Springer, 2007. v. 18.
- FARIA, Daniel; PESQUITA, Catia; SANTOS, Emanuel; PALMONARI, Matteo; CRUZ, Isabel F; COUTO, Francisco M. The agreementmakerlight ontology matching system. In: SPRINGER. **OTM Confederated International Conferences"On the Move to Meaningful Internet Systems"**. [S.l.], 2013. p. 527–541.
- FERRANTI, Nicolas; MOURO, Jose Ronaldo; MENDONÇA, Fabricio Martins; SOUZA, Jairo Francisco de; SOARES, Stenio Sa Rosario Furtado. A framework for evaluating ontology meta-matching approaches. **Journal of Intelligent Information Systems**, Springer, 2020.
- FERRANTI, Nicolas; SOARES, Stênio Sã Rosário Furtado; SOUZA, Jairo Francisco de. Combining meta-heuristics and linear programming to address ontology meta-matching problem. In: **SBBD Companion**. [S.l.: s.n.], 2018. p. 47–52.
- FERRANTI, Nicolas; SOARES, Stênio Sa Rosário Furtado; SOUZA, Jairo Francisco de. Meta-alinhamento de ontologias utilizando a abordagem presa-predador. In: **SBBD**. [S.l.: s.n.], 2018. p. 49–60.
- FORSATI, Rana; SHAMSFARD, Mehrnoush. Symbiosis of evolutionary and combinatorial ontology mapping approaches. **Information Sciences**, Elsevier, v. 342, p. 53–80, 2016.
- FORTNOW, Lance et al. Time for computer science to grow up. **Communications of the ACM**, v. 52, n. 8, p. 33–35, 2009.
- FREIRE, Juliana; KOOP, David; SANTOS, Emanuele; SILVA, Cláudio T. Provenance for computational tasks: A survey. **Computing in Science & Engineering**, IEEE, v. 10, n. 3, 2008.
- GARG, Poonam. A comparison between memetic algorithm and genetic algorithm for the cryptanalysis of simplified data encryption standard algorithm. **International Journal of Network Security & Its Applications**, 2010.
- GEEM, Zong Woo; KIM, Joong Hoon; LOGANATHAN, Gobichettipalayam Vasudevan. A new heuristic optimization algorithm: harmony search. **simulation**, Sage Publications Sage CA: Thousand Oaks, CA, v. 76, n. 2, p. 60–68, 2001.

GRUBER, Thomas R. A translation approach to portable ontology specifications. **Knowledge Acquisition**, v. 5, n. 2, p. 199–220, 1993.

GRUBER, Thomas R. A translation approach to portable ontology specifications. **Knowledge acquisition**, Elsevier, v. 5, n. 2, p. 199–220, 1993.

GULIĆ, Marko; VRDOLJAK, Boris; PTIČEK, Marina. Automatically specifying a parallel composition of matchers in ontology matching process by using genetic algorithm. **Information**, Multidisciplinary Digital Publishing Institute, v. 9, n. 6, p. 138, 2018.

HERTLING, Sven; PORTISCH, Jan; PAULHEIM, Heiko. Melt-matching evaluation toolkit. In: SPRINGER. **International Conference on Semantic Systems**. [S.l.], 2019. p. 231–245.

HOEHNDORF, Robert; SCHOFIELD, Paul N; GKOUTOS, Georgios V. The role of ontologies in biological and biomedical research: a functional perspective. **Briefings in bioinformatics**, Oxford University Press, v. 16, n. 6, p. 1069–1080, 2015.

JIANG, Li; XUE, Xingsi. Optimizing ontology alignments through nsga-ii using an aggregation strategy and a mapping extraction approach. In: IEEE. **2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)**. [S.l.], 2015. p. 349–352.

JIANG, L.; XUE, Xingsi; TSAI, Pei-Wei; PAN, Jeng-Shyang. A novel ontology matching technology based on nsga-ii. **Journal of Information Hiding and Multimedia Signal Processing**, v. 7, p. 317–324, 01 2016.

KEELE, Staffs et al. **Guidelines for performing systematic literature reviews in software engineering**. [S.l.], 2007.

KESSENTINI, Marouane; OUNI, Ali; LANGER, Philip; WIMMER, Manuel; BECHIKH, Slim. Search-based metamodel matching with structural and syntactic measures. **Journal of Systems and Software**, Elsevier, v. 97, p. 1–14, 2014.

KIM, Jinseok. Author-based analysis of conference versus journal publication in computer science. **Journal of the Association for Information Science and Technology**, Wiley Online Library, v. 70, n. 1, p. 71–82, 2019.

KUHN, Harold W. The hungarian method for the assignment problem. **Naval research logistics quarterly**, Wiley Online Library, v. 2, n. 1-2, p. 83–97, 1955.

KUREYCHIK, Viktor; SEMENOVA, Alexandra. Combined method for integration of heterogeneous ontology models for big data processing and analysis. In: SPRINGER. **Computer Science On-line Conference**. [S.l.], 2017. p. 302–311.

LAUMANN, Marco; RUDOLPH, Günter; SCHWEFEL, Hans-Paul. A spatial predator-prey approach to multi-objective optimization: A preliminary study. In: SPRINGER. **Parallel Problem Solving from Nature—PPSN V**. [S.l.], 1998. p. 241–249.

LEACOCK, Claudia; CHODOROW, Martin. Combining local context and wordnet similarity for word sense identification. **WordNet: An electronic lexical database**, v. 49, n. 2, p. 265–283, 1998.

LEBO, Timothy; SAHOO, Satya; MCGUINNESS, Deborah; BELHAJJAME, Khalid; CHENEY, James; CORSAR, David; GARIJO, Daniel; SOILAND-REYES, Stian; ZEDNIK, Stephan; ZHAO, Jun. Prov-o: The prov ontology. **W3C recommendation**, W3C, v. 30, 2013.

LEVENSHTEIN, Vladimir I. Binary codes capable of correcting deletions, insertions, and reversals. In: **Soviet physics doklady**. [S.l.: s.n.], 1966. v. 10, n. 8, p. 707–710.

LIM, Chunhyeok; LU, Shiyong; CHEBOTKO, Artem; FOTOUHI, Farshad. Prospective and retrospective provenance collection in scientific workflow environments. In: **IEEE. 2010 IEEE International Conference on Services Computing**. [S.l.], 2010. p. 449–456.

MAEDCHE, Alexander; STAAB, Steffen. Measuring similarity between ontologies. In: **SPRINGER. International Conference on Knowledge Engineering and Knowledge Management**. [S.l.], 2002. p. 251–263.

MANNING, Christopher D.; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich. **Introduction to Information Retrieval**. USA: Cambridge University Press, 2008. ISBN 0521865719.

MARJIT, Ujjal. Aggregated similarity optimization in ontology alignment through multiobjective particle swarm optimization. **International Journal of Advanced Research in Computer and Communication Engineering**, v. 4, n. 2, 2015.

MARTINEZ-GIL, Jorge; ALDANA-MONTES, José F. An overview of current ontology meta-matching solutions. **The Knowledge Engineering Review**, Cambridge University Press, v. 27, n. 4, p. 393–412, 2012.

MARTINEZ-GIL, Jorge; NAVAS-DELGADO, Ismael; ALDANA-MONTES, José F. Maf: An ontology matching framework. **Journal of Universal Computer Science**, v. 18, n. 2, p. 194–217, 2012.

MARTÍNEZ-ROMERO, Marcos; VÁZQUEZ-NAYA, José Manuel; NÓVOA, Francisco Javier; VÁZQUEZ, Guillermo; PEREIRA, Javier. A genetic algorithms-based approach for optimizing similarity aggregation in ontology matching. In: **SPRINGER. International Work-Conference on Artificial Neural Networks**. [S.l.], 2013. p. 435–444.

MCBRIDE, Brian. Jena: A semantic web toolkit. **IEEE Internet computing**, IEEE, v. 6, n. 6, p. 55–59, 2002.

MELNIK, Sergey; GARCIA-MOLINA, Hector; RAHM, Erhard. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: **IEEE. Proceedings 18th International Conference on Data Engineering**. [S.l.], 2002. p. 117–128.

METROPOLIS, Nicholas; ROSENBLUTH, Arianna W; ROSENBLUTH, Marshall N; TELLER, Augusta H; TELLER, Edward. Equation of state calculations by fast computing machines. **The journal of chemical physics**, AIP, v. 21, n. 6, p. 1087–1092, 1953.

MEYER, Bertrand; CHOPPY, Christine; STAUNSTRUP, Jørgen; LEEUWEN, Jan Van. Research evaluation for computer science. **Communications of the ACM**, v. 52, n. 4, p. 31–34, 2009.

MOHAMMADI, Majid; HOFMAN, Wout; TAN, Yao-Hua. A comparative study of ontology matching systems via inferential statistics. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 31, n. 4, p. 615–628, 2018.

MOHAMMADI, Majid; HOFMAN, Wout; TAN, Yao-Hua. Simulated annealing-based ontology matching. **ACM Transactions on Management Information Systems (TMIS)**, ACM, v. 10, n. 1, p. 3, 2019.

NGO, DuyHoa; BELLAHSENE, Zohra. Overview of yam++—(not) yet another matcher for ontology alignment task. **Web Semantics: Science, Services and Agents on the World Wide Web**, Elsevier, v. 41, p. 30–49, 2016.

OCHIENG, Peter; KYANDA, Swaib. Large-scale ontology matching: State-of-the-art analysis. **ACM Computing Surveys (CSUR)**, ACM, v. 51, n. 4, p. 75, 2018.

OTERO-CERDEIRA, Lorena; RODRÍGUEZ-MARTÍNEZ, Francisco J; GÓMEZ-RODRÍGUEZ, Alma. Ontology matching: A literature review. **Expert Systems with Applications**, Elsevier, v. 42, n. 2, p. 949–971, 2015.

PATTERSON, David; SNYDER, Lawrence; ULLMAN, Jeffrey. Evaluating computer scientists and engineers for promotion and tenure. **Computing Research News**, 1999.

PAULHEIM, Heiko. Evaluating ontology matchers on real-world financial services data models. 2019.

PETERSEN, Kai; FELDT, Robert; MUJTABA, Shahid; MATTSSON, Michael. Systematic mapping studies in software engineering. In: **Ease**. [S.l.: s.n.], 2008. v. 8, p. 68–77.

POLI, Riccardo; LANGDON, William B; MCPHEE, Nicholas F; KOZA, John R. **A field guide to genetic programming**. [S.l.]: Lulu. com, 2008.

RAMESH, M; KARTHIKEYAN, India P; MEENACHI, India Dr N Madurai; BABA, M Sai. Optimizing ontology alignment for nuclear information system. **International Journal of Emerging Technologies in Engineering Research**, v. 4, 2016.

RANA, Vijay; SINGH, Gurdev. Analysis of ontology matching system and their countermeasures. In: IEEE. **2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)**. [S.l.], 2014. p. 83–88.

RESNIK, Philip. Using information content to evaluate semantic similarity in a taxonomy. In: **Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995. (IJCAI'95), p. 448–453. ISBN 1558603638.

ROUCES, Jacobo; MELO, Gerard de; HOSE, Katja. Complex schema mapping and linking data: Beyond binary predicates. In: **LDOW@ WWW**. [S.l.: s.n.], 2016.

SARASA, Guillermo; GRANADOS, Ana; RODRIGUEZ, Francisco B. Algorithmic clustering based on string compression to extract p300 structure in eeg signals. **Computer methods and programs in biomedicine**, Elsevier, v. 176, p. 225–235, 2019.

SEMENOVA, Alexandra; KUREYCHIK, Viktor. Application of swarm intelligence for domain ontology alignment. In: SPRINGER. **Proceedings of the First International Scientific Conference “Intelligent Information Technologies for Industry”(IITI'16)**. [S.l.], 2016. p. 261–270.

SEMENOVA, AV; KUREYCHIK, VM. Multi-objective particle swarm optimization for ontology alignment. In: IEEE. **2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT)**. [S.l.], 2016. p. 1–7.

SHI, Yuhui; EBERHART, Russell. A modified particle swarm optimizer. In: IEEE. **1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)**. [S.l.], 1998. p. 69–73.

SHVAIKO, Pavel; EUZENAT, Jérôme. Ontology matching: state of the art and future challenges. **IEEE Transactions on knowledge and data engineering**, IEEE, v. 25, n. 1, p. 158–176, 2013.

SILVA, Mateus Ferreira; BAIÃO, Fernanda Araujo; REVOREDO, Kate. Towards planning scientific experiments through declarative model discovery in provenance data. In: IEEE. **2014 IEEE 10th International Conference on e-Science**. [S.l.], 2014. v. 2, p. 95–98.

SINGH, Aarti. Optimizing ontology mapping using genetic algorithms (oomga). **International Journal of Advanced Networking and Applications**, Eswar Publications, v. 9, n. 5, p. 3571–3579, 2018.

SOUZA, Jairo Francisco de; SIQUEIRA, Sean Wolfgang Matsui; NUNES, Bernardo. A framework to aggregate multiple ontology matchers. **International Journal of Web Information Systems**, Emerald Publishing Limited, 2019.

STEINMACHER, Igor; CHAVES, Ana Paula; GEROSA, Marco Aurélio. Awareness support in distributed software development: A systematic review and mapping of the literature. **Computer Supported Cooperative Work (CSCW)**, Springer, v. 22, n. 2-3, p. 113–158, 2013.

STOILLOS, Giorgos; STAMOU, Giorgos; KOLLIAS, Stefanos. A string metric for ontology alignment. In: SPRINGER. **International Semantic Web Conference**. [S.l.], 2005. p. 624–637.

TALBI, El-Ghazali. **Metaheuristics: from design to implementation**. [S.l.]: John Wiley & Sons, 2009. v. 74.

THIÉBLIN, Elodie; HAEMMERLÉ, Ollivier; HERNANDEZ, Nathalie; TROJAHN, Cassia. Survey on complex ontology matching. **Semantic Web**, IOS Press, n. Preprint, p. 1–39, 2019.

TILAHUN, Surafel Lulseged; ONG, Hong Choon. Prey-predator algorithm: A new metaheuristic algorithm for optimization problems. **International Journal of Information Technology & Decision Making**, World Scientific, v. 14, n. 06, p. 1331–1352, 2015.

VARDI, Moshe Y. Revisiting the publication culture in computing research. **Communications of the ACM**, ACM, v. 53, n. 3, p. 5–5, 2010.

VRETTAS, George; SANDERSON, Mark. Conferences versus journals in computer science. **Journal of the Association for Information Science and Technology**, Wiley Online Library, v. 66, n. 12, p. 2674–2684, 2015.

WINKLER, William E. The state of record linkage and current research problems. In: CITESEER. **Statistical Research Division, US Census Bureau**. [S.l.], 1999.

WU, Zhibiao; PALMER, Martha. Verbs semantics and lexical selection. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 32nd annual meeting on Association for Computational Linguistics**. [S.l.], 1994. p. 133–138.

XUE, Xingsi. Efficient ontology meta-matching using alignment prescreening approach and gaussian random field model assisted nsga-ii. **Journal of Information Hiding and Multimedia Signal Processing**, v. 8, p. 1062–1068, 01 2017.

XUE, Xingsi; CHEN, Junfeng. A preference-based multi-objective evolutionary algorithm for semiautomatic sensor ontology matching. **International Journal of Swarm Intelligence Research (IJSIR)**, IGI Global, v. 9, n. 2, p. 1–14, 2018.

XUE, Xingsi; CHEN, Junfeng. Matching biomedical ontologies through compact differential evolution algorithm. **Systems Science & Control Engineering**, Taylor & Francis, v. 7, n. 2, p. 85–89, 2019.

XUE, Xingsi; CHEN, Junfeng. Optimizing ontology alignment through hybrid population-based incremental learning algorithm. **Memetic Computing**, Springer, v. 11, n. 2, p. 209–217, 2019.

XUE, Xingsi; CHEN, Junfeng. Using compact evolutionary tabu search algorithm for matching sensor ontologies. **Swarm and Evolutionary Computation**, Elsevier, v. 48, p. 25–30, 2019.

XUE, X.; CHEN, J.; CHEN, D. Matching biomedical ontologies through compact hybrid evolutionary algorithm. **Journal of Information Hiding and Multimedia Signal Processing**, v. 10, p. 110–117, 01 2019.

XUE, Xingsi; CHEN, Jie; CHEN, Junfeng; CHEN, Dongxu. A hybrid nsga-ii for matching biomedical ontology. In: SPRINGER. **International Conference on Intelligent Information Hiding and Multimedia Signal Processing**. [S.l.], 2018. p. 3–10.

XUE, Xingsi; CHEN, Jie; CHEN, Junfeng; CHEN, Dongxu. Using compact coevolutionary algorithm for matching biomedical ontologies. **Computational intelligence and neuroscience**, Hindawi, v. 2018, 2018.

XUE, Xingsi; CHEN, Jie; LIU, Jianhua; CHEN, Dongxu. Matching biomedical ontologies through compact evolutionary simulated annealing algorithm. In: SPRINGER. **International Conference on Genetic and Evolutionary Computing**. [S.l.], 2018. p. 661–668.

XUE, Xingsi; CHEN, Junfeng; YAO, Xin. Efficient user involvement in semiautomatic ontology matching. **IEEE Transactions on Emerging Topics in Computational Intelligence**, IEEE, 2018.

XUE, Xingsi; HANG, Zhi; TANG, Zhengyi. Interactive biomedical ontology matching. **PLoS one**, Public Library of Science, v. 14, n. 4, p. e0215147, 2019.

XUE, Xingsi; LIU, Jianhua. Collaborative ontology matching based on compact interactive evolutionary algorithm. **Knowledge-Based Systems**, Elsevier, v. 137, p. 94–103, 2017.

XUE, Xingsi; LIU, Jianhua. A compact hybrid evolutionary algorithm for large scale instance matching in linked open data cloud. **International Journal on Artificial Intelligence Tools**, World Scientific, v. 26, n. 04, p. 1750013, 2017.

XUE, Xingsi; LIU, Jianhua. Optimizing ontology alignment through compact moea/d. **International Journal of Pattern Recognition and Artificial Intelligence**, World Scientific, v. 31, n. 04, p. 1759004, 2017.

- XUE, Xingsi; LIU, Jianhua. Geo-spatial ontology matching through compact evolutionary algorithm. In: SPRINGER. **International Conference on Smart Vehicular Technology, Transportation, Communication and Applications**. [S.l.], 2018. p. 11–18.
- XUE, Xingsi; LIU, Jianhua; TSAI, Pei-Wei; ZHAN, Xianyin; REN, Aihong. Optimizing ontology alignment by using compact genetic algorithm. In: IEEE. **2015 11th International Conference on Computational Intelligence and Security (CIS)**. [S.l.], 2015. p. 231–234.
- XUE, Xingsi; LIU, Shijian; WANG, Jinshui. A hybrid central force optimization algorithm for optimizing ontology alignment. In: SPRINGER. **The Euro-China Conference on Intelligent Data Analysis and Applications**. [S.l.], 2017. p. 334–343.
- XUE, Xingsi; LU, Jiawei; CHEN, Junfeng. Using nsga-iii for optimising biomedical ontology alignment. **CAAI Transactions on Intelligence Technology**, IET, 2019.
- XUE, Xingsi; PAN, Jeng-Shyang. A segment-based approach for large-scale ontology matching. **Knowledge and Information Systems**, Springer, v. 52, n. 2, p. 467–484, 2017.
- XUE, Xingsi; PAN, Jeng-Shyang. A compact co-evolutionary algorithm for sensor ontology meta-matching. **Knowledge and Information Systems**, Springer, v. 56, n. 2, p. 335–353, 2018.
- XUE, Xingsi; REN, Aihong. A large scale multi-objective ontology matching framework. In: SPRINGER. **International Conference on Intelligent Information Hiding and Multimedia Signal Processing**. [S.l.], 2017. p. 250–255.
- XUE, Xingsi; TANG, Zhengyi. An evolutionary algorithm based ontology matching system. **J. Inf. Hiding Multimed. Signal Process**, v. 8, n. 3, p. 551–556, 2017.
- XUE, Xingsi; TSAI, Pei-Wei; FENG, Guirong. Efficient ontology meta-matching based on metamodel-assisted compact moea/d. **Journal of Information Hiding and Multimedia Signal Processing**, v. 8, 2017.
- XUE, Xingsi; TSAI, Pei-Wei; WANG, Jinshui. Using compact memetic algorithm for optimizing ontology alignment. **ICIC Express Letters**, v. 11, p. 53–58, 01 2017.
- XUE, Xingsi; TSAI, Pei-Wei; ZHANG, Li-Li. Using parallel compact evolutionary algorithm for optimizing ontology alignment. In: SPRINGER. **International Conference on Genetic and Evolutionary Computing**. [S.l.], 2016. p. 157–165.
- XUE, Xingsi; WANG, Jinshui; REN, Aihong. A novel system architecture for ontology matching. In: IEEE. **2016 12th International Conference on Computational Intelligence and Security (CIS)**. [S.l.], 2016. p. 100–102.
- XUE, Xingsi; WANG, Yuping. Ontology alignment based on instance using nsga-ii. **Journal of Information Science**, Sage Publications Sage UK: London, England, v. 41, n. 1, p. 58–70, 2015.
- XUE, Xingsi; WANG, Yuping. Optimizing ontology alignments through a memetic algorithm using both matchmeasure and unanimous improvement ratio. **Artificial Intelligence**, Elsevier, v. 223, p. 65–81, 2015.
- XUE, Xingsi; WANG, Yuping. Using memetic algorithm for instance coreference resolution. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 28, n. 2, p. 580–591, 2015.

XUE, Xingsi; WANG, Yuping. Improving the efficiency of nsga-ii based ontology aligning technology. **Data & Knowledge Engineering**, Elsevier, v. 108, p. 1–14, 2017.

XUE, Xingsi; WANG, Yuping; HAO, Weichen. Using moea/d for optimizing ontology alignments. **Soft Computing**, Springer, v. 18, n. 8, p. 1589–1601, 2014.

XUE, Xingsi; WANG, Yuping; HAO, Weichen. Optimizing ontology alignments by using nsga-ii. **International Arab Journal of Information Technology (IAJIT)**, v. 12, n. 2, 2015.

XUE, Xingsi; WANG, Yuping; HAO, Weichen; HOU, Juan. Optimizing ontology alignments through nsga-ii without using reference alignment. **Computing and Informatics**, v. 33, n. 4, p. 857–876, 2015.

XUE, Xingsi; WANG, Yuping; REN, Aihong. Optimizing ontology alignment through memetic algorithm based on partial reference alignment. **Expert Systems with Applications**, Elsevier, v. 41, n. 7, p. 3213–3222, 2014.

Yager, R. R. On ordered weighted averaging aggregation operators in multicriteria decision-making. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 18, n. 1, p. 183–190, 1988.

YU, Xin-mei; XIONG, Xin-yin; WU, Yao-wu. A pso-based approach to optimal capacitor placement with harmonic distortion consideration. **Electric Power Systems Research**, Elsevier, v. 71, n. 1, p. 27–33, 2004.

ZHANG, Feng; GUO, Zongliang. A novel optimization method for ontology matching based on heuristic population evolution algorithm. **Arabian Journal for Science and Engineering**, Springer, v. 44, n. 4, p. 3137–3153, 2019.