Data Analysis on Effective Strategies to Enhance the Semantics of OWL Ontologies

Tibet Tugay
https://orcid.org/0009-0007-8315-9760
University of Twente
The Netherlands

ABSTRACT

OWL has been created to support the development of the Semantic Web, a prime initiative that aims to make the Web more machine-readable and actionable. Many believe that the ontologies created with OWL can be further improved, especially in regards to the interoperability and reusability aspects of the FAIR principles. To this end, gUFO, a lightweight implementation of UFO, can better support the semantics of the OWL ontologies to ensure the FAIR principles are incorporated into the generated models, reducing the possibility of having interoperability problems such as the false agreement problem. This work aimed to contribute to enhancing the semantics of OWL ontologies in the most effective and automated way by analyzing the usage of OWL meta-properties in Scior. Scior is a software that infers the ontological categories of OWL classes through gUFO. The objective of this work was to look for better practices for this process to identify the best initial seeding. Throughout the data analysis, 8 strategies related to the position, sortality, and rigidity of the initial seeding have been examined. The results and the subsequent discussion of them suggest that there are indeed better strategies to adopt while enhancing the semantics of OWL ontologies with Scior.

KEYWORDS

Enhancing Semantics, OWL, Scior, gUFO, FAIR Principles

1 INTRODUCTION

The Semantic Web is a prime initiative that aims to make the Web more machine-readable and actionable. To this end, World Wide Web Consortium (W3C) has created the Web Ontology Language (OWL) [10].

Many believe that the ontologies created with OWL can be further improved, especially in regards to the interoperability and reusability aspects of the FAIR (**F**indable, **A**ccessible, **I**nteroperable, and **R**eusable) principles (further explained in Section 1.1.1) [6]. This work analyzed one of the more recent ways to enhance the semantics of OWL ontologies and pursue the most effective strategies to use while doing so.

TScIT 41, July 5, 2024, Enschede, The Netherlands

1.1 Contextualization

The Semantic Web is the idea of a second-generation web pioneered by W3C. It "is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web" [10]. The Semantic Web aims to facilitate this automatic processing of varied sections of information about a given object, even if it is stored in segmented web resources [15]. OWL has been developed to aid the need for machine-processable conceptualizations that arise with the idea of the Semantic Web, expediting the ability of a machine to understand not only the meaning of the words about an object but also the varied concepts and knowledge surrounding it [10].

Even though OWL has ontology in its name, it is still a logical language without an explicit commitment to a Foundational Ontology. OWL can act as a meaning contract, capturing the conceptualizations. However, due to the absence of a commitment to a Foundational Ontology, it lacks "formal methods and theories for clarifying conceptualizations and articulating their representations" [6]. This results in OWL ontologies prioritizing the content of the ontology over the expressivity about the meta-properties of the said content. Therefore, OWL may not have inherent guidelines to generate FAIR (further explained in Section 1.1.1) models in the language itself [6].

In contrast to OWL, OntoUML, a Unified Foundational Ontology-based (UFO) conceptual modeling language, has inherent guidelines to encourage easier generation FAIR ontologies. OntoUML is truly ontological due to the ontological distinctions of UFO. OntoUML, assuming there is no human error involved, does not suffer from problems, e.g. the false agreement problem, that can compromise the interoperability and reusability of the OWL language [6].

Alongside OntoUML, gUFO [2] is a lightweight implementation of UFO, that allows OWL ontologies to partially use the ontological distinctions and logical constraints of UFO [4]. Thus, gUFO can better support the semantics of the OWL ontologies to ensure the FAIR principles (further explained in Section 1.1.1) are incorporated into the generated models.

1.1.1 FAIR Principles. The FAIR principles were created in 2016 and, ever since then, have been endorsed and adopted by many independent stakeholders [8]. The principles aim to mainstream good data stewardship practices to generate more machine-actionable data [8]. In other words, they aim to make the data more **F**indable, **A**ccessible, **I**nteroperable, and **R**eusable to machines and humans.

The main reason for enhancing the semantics of OWL ontologies is to make them more interoperable and reusable so that the semantic web applications can use good quality data and have an agreement for the collective understanding of it. Therefore, to

Tibet Tugay
https://orcid.org/0009-0007-8315-9760

comply with the main goals of this research, this work aimed to follow the FAIR principles as much as possible while generating its deliverables.

1.2 Motivation

One major problem the lack of FAIR principles creates for an ontology is the false agreement problem, which arises from two models falsely thinking that they agree on the data they conceptualize and how they conceptualize it [6]. This hinders interoperability between the two models, as they think they conceptualize the same concepts, but in actuality, they do not.

To explain how the false agreement problem can cause issues in real-world applications, an organ transplant example from [6] has been provided. As explained in its original paper, even though Transplant-A (transplant class in model A) and Transplant-B (transplant class in model B) are both transplants, Transplant-A is the individual transplants that happen in a given space and time while Transplant-B is the types of transplants a surgeon can perform. Thus, Transplant-A is an instantiation of Transplant-B, not an identity [6].

In the context of the semantic web, interoperability and reusability problems obstruct the correct representation of the real world in the constructed ontologies. Therefore, decreasing the quality of the understanding generated or data collected by the machines operating in the semantic web. To achieve a cleaner operation of machines on the ontologies, the following proposal has been developed.

1.3 Proposal

The objective of this work is to contribute to the enhancement of the semantics of OWL ontologies in the most effective and automated way. To do so, the usage of OWL meta-properties will be analyzed in a software that infers the ontological categories of OWL classes through gUFO, looking for better practices for this process to identify the best initial seeding. This will result in better classification of the concepts and more suitable constraints throughout the ontology, enhancing the semantics of the ontology. Thus making it more expressive and reusable.

1.4 Research Questions

To realize the goals of this work, the following questions have been considered:

- **Research Question**: What are the best seeding strategies for a software tool that performs automatic inference of ontological categories to OWL ontologies to enhance their semantics?
- **Sub-Question 1**: What are the patterns that result in the most effective initial seeding?
- **Sub-Question 2**: What are the scenarios that decrease the quality of the most effective seeding drastically?

The rest of this work will discuss the subject at hand in the subsequent manner: Section 2 will go over the technical background information necessary to follow the paper. Section 3 will argue the novelty of this work through the related works. Section 4 will present the problem statement and a plan to tackle the problem statement as well as the tools and techniques that will be used during this process. Section 5 will display the data analysis results and how to interpret them effectively. Section 6 will examine the results and confer them to the metric of this work. Lastly, Section 7 will review the efforts of this work and any suggested future work.

2 TECHNICAL BACKGROUND

The lightweight implementation of UFO, named gUFO, focuses on endurant types. Endurant types are types of classes that are object-like, meaning they can change their qualities, like how an object changes through time, and endure in time [4]. The classifications and the types for Endurant types in gUFO are taken from UFO. Thus Figure 1 serves to show the gUFO types as well as the UFO ones.

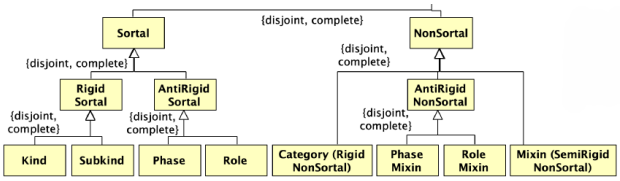

Figure 1: Part of UFO Taxonomy for Endurant Types [7]

As seen from Figure 1, endurant types have two meta-properties that allow them to be classified into their class types: *Rigidity* and *Sortality*. *Rigidity* is a class' ability to change their class. *Rigid* classes cannot change their class throughout their existence while *Anti-Rigid* classes can change theirs [4]. For example, a person cannot cease to be a person while a student can graduate and discontinue being a student. These two classes are a *Rigid* and an *Anti-Rigid* respectively. Moreover, a class can also be a *Semi-Rigid*, meaning some of their instances are *Rigid* and others are *Anti-Rigid* [4]. The *Anti-Rigids* and *Semi-Rigids* have been collected under the umbrella abstract class of *Non-Rigid*, as they both do not conform to the *Rigidity* of a class.

On the other hand, the *Sortality* of a class is tied to the principle of identity of a class. A principle of identity is an idea the class represents. In a more formal manner, a "principle of identity makes explicit the properties that no two individuals can have in common, since such properties uniquely identify them" [4]. A *Sortal* class is a class that has a single principle of identity, while *Non-Sortal* classes represent the commonality of different principles of identity [4]. An example for a *Sortal* class would be a person as a person is composed of the idea of a person. Meanwhile, an example for a *Non-Sortal* class would be a work of art, as a work of art is composed of the idea of music, paintings, sculptures, etc.

The *Sortality* and *Rigidity* of a class will be significant when the initial seeding strategy is discussed. The initial seeding is the only class that Scior learns the information of before running it on the rest of the taxonomy, namely a subset of the model with no domain associations [3]. The initial seeding allows Scior to extract information about the rest of the taxonomy by using its inference rules, as mentioned in [4].

Another property that will be important while discussing the initial seeding strategy is the *Position* of the initial seeding within the taxonomy. A taxonomy can be seen as a hierarchical representation of the classes in a subset of the model [3]. Due to this hierarchy, the nodes in the taxonomy follow a tree-like structure. The *Root* nodes are at the start of the taxonomy, meaning they do not have

any superclasses. On the contrary, the *Leaf* nodes are at the ends of the taxonomy, meaning they do not have any subclasses. Lastly, *Intermediate* nodes are all the nodes that are in between the *Roots* and the *Leaves*.

Furthermore, it is important to clearly define some more terminology that will be useful while discussing the results of this work. There are 8 types an endurant type class can have in gUFO. These are shown as the leaves in Figure 1. However, there are 14 classifications an endurant type class can have. The classifications include the 8 types and all the 6 meta-property options it has corresponding to their *Rigidity* and *Sortality*.

Lastly, there are two assumptions that will be relevant to the results of this work: Closed-World Assumption (CWA) and Open-World Assumption (OWA). These are also the two modes of running Scior on the Catalog. When Scior runs in CWA mode, it assumes the ontology is fully declared, meaning that any class and properties related to the ontology is processed by Scior [4]. In contrast, when it is in OWA mode, it assumes that some of the information regarding the classes and properties of an ontology can be missing.

3 RELATED WORKS

There has not been any data analysis study on enhancing the semantics of OWL ontologies through gUFO, since gUFO is a recent development. However, there have been other works on improving the semantics of OWL through other means, mostly focusing on creating new tools, instead of analyzing them.

The most relevant work preceding this one is Analysis of the OWL ontologies [5]. It is an analysis of the following OWL enhancement tools: ONTOMETRIC, OntoQA, and Protégé [5]. There are two main differences between [5] and this work. The first one is the difference in the tools that are being analyzed. The Analysis of the OWL ontologies analyzes the aforementioned tools while this work has only analyzed Scior. The second is the purpose of the analysis. The Analysis of the OWL ontologies analyzes the 3 tools to “normalize the ontology metrics as a pre-process to apply structural metrics" [5]. In contrast, this work will analyze Scior’s initial seeding selection to find the most effective way to automate the application of gUFO meta-properties to OWL ontologies, enhancing the ontologies.

Another work that focuses on enhancing OWL is the OWL Enhance prototype [9], but it has considerable contrasts with the objective of this work. The OWL Enhance prototype focuses on enhancing only the relation semantics of OWL and does that by eliciting knowledge from the providers [9]. Compared to this approach, the automated application of gUFO to OWL meta-properties is unique.

In addition to OWL Enhance prototype, the Description Logic Entailment-Based (DLE) OWL Reasoning is another tool that has the aim of improving OWL semantics [11]. It uses description logic reasoners and artificial intelligence to enhance entailment-based OWL reasoning [11]. However, neither description logic reasoners nor entailment-based reasoning are within the objective or scope of this work.

Conversely, OWL2Jess is a prototype that aims to enhance OWL semantics by enhancing the reasoning of OWL ontologies [12]. Once again, even though enhancing the semantics is a common objective, OWL2Jess reaches that objective through enhancing reasoning, while this work will analyze enhancing the expressivity.

Lastly, a relevant study does not focus on enhancing OWL semantics but focuses instead on predicting OntoUML stereotypes. [1] discusses the idea of training a Graph Neural Network (GNN) to predict meta-classes of Unified Modeling Language (UML) class diagrams to automate stereotyping of legacy models. This is achieved by transforming OntoUML models into Conceptual Knowledge Graphs and feeding them into the GNN. This work is distinct compared to [1] since it does not focus on AI training and focuses on enhancing the semantics of OWL ontologies instead of predicting OntoUML meta-classes.

As can be seen from the explanations of the aforementioned tools, this work aims to achieve something novel compared to all of them, since it does not focus on creating a tool. The only analysis done on any tool that focuses on increasing the semantics of OWL focuses on normalizing ontological and structural metrics [5]. Instead, this work will aspire to enhance OWL semantics by the automated application of gUFO to OWL meta-properties and analyzing the most effective ways to do so. How this has been done has been discussed in the next section.

4 METHODOLOGY

This work aimed to examine enhancing the semantics of OWL ontologies through the language’s relations with UFO, OntoUML, and gUFO. The relations of OWL to these concepts can be seen in Figure 2. The taxonomies of the OntoUML models were mapped into OWL taxonomies. The taxonomies have been enhanced with Scior, with the usage of gUFO, by putting different meta-properties in the initial seeding to find the most effective initial seeding. Furthermore, the effectiveness of these enhancements, relative to each other, has been discussed through data analysis to extract any patterns that might be relevant to the effectiveness of different initial seedings.

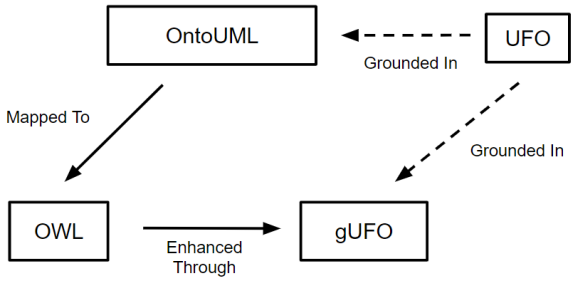

Figure 2: OWL Relations

4.1 Tools

The two major tools used in the methodology are Scior [4] and the OntoUML/UFO Catalog (Catalog) [3]. The Catalog is a “structured, collaborative, and open-source catalog that contains UFO-grounded models" [3]. The vast majority of the Catalog models are represented in OntoUML, and the OntoUML models can be mapped into an OWL ontology, keeping only their classes and generalizations to serve the purpose of this work [4].

The objective of Scior is "to infer the UFO meta-categories of classes in an OWL ontology given an initial seeding" [4]. This initial seeding is the real gUFO type of a singular class. In its automatic setting, Scior takes this initial seeding and outputs what gUFO type each class can be alongside the gUFO classifications (such as *Sortality* and *Rigidity*) of each class, resulting in a clearer understanding of each class through the usage of gUFO. This can also be accomplished in a non-automatic manner where Scior presents the user options to pick from. For the purposes of this work, Scior has only been used in its automatic function.

4.1.1 Scior Tester. The environment to run Scior on the data from the Catalog has been implemented with the help of Scior Tester [14]. Scior Tester is the environment created to test the correctness and effectiveness of Scior from its original paper [4].

The `build` function of Scior Tester has extracted the taxonomies from the OWL ontologies mapped from the Catalog, just like it was done in the original paper [4].

After that, `Test1` has been run for all the resulting taxonomies. `Test1` takes the following steps:

- Strips all the stereotypes from the classes in a given taxonomy.
- Chooses one class and marks it as the initial seeding for the next run of Scior.
- Runs Scior with the initial seeding.
- Removes the initial seeding and chooses a new class to mark as initial seeding.
- Moves on to the next taxonomy when all the classes in a taxonomy have been marked as the initial seeding once.

Marking a class as the initial seeding is done by taking its true value from the stereotype and labeling the class with the said type before running Scior on the whole taxonomy.

4.1.2 Scior Dataset. Due to the bugs and the compatibility issues between Scior and the current Scior Tester, the Scior Tester was not able to generate the necessary data from the Catalog. Instead, the dataset generated for the original paper, by the use of the older Scior Tester that was fully operational at the time of the original paper, was used.

There are advantages and disadvantages to using the Scior Dataset [13] instead of generating new data from the Catalog with the Scior Tester. These advantages and disadvantages are as follows:

Advantages:

- The data in Scior Dataset is peer-reviewed. Thus, it is a lot more reliable than creating another dataset with the tools provided.
- Scior Dataset is already able to be used for data analysis compared to the current problems with the Scior Tester.

Disadvantages:

- The Catalog has been updated with more models since the creation of the original paper and the Scior Dataset. Thus, there will be fever data to analyze than what the current Catalog holds.

Furthermore, there was an assortment of taxonomies in Scior Dataset that were not used in the data analysis. One of the reasons for this was that some taxonomies lack all the required files to be analyzed properly, presumably due to human error at some point along the creation of the Scior Dataset. The list of taxonomies that were skipped due to the lack of all required files can be found in the data analysis repository [16] under the documentation directory.

Another reason to skip some taxonomies was the size of the taxonomies. A considerable amount of taxonomies have less than 5 classes in them. Less than 5 classes disturb the percentage values used during the data analysis as even a minute change in a small taxonomy has a huge impact on the percentage values (as discussed in Section 5). The list of taxonomies that were skipped due to having less than 5 classes in them can be found in the data analysis repository [16] under the documentation directory as well. Furthermore, the skipped taxonomies have been skipped for both CWA and OWA environments.

The Scior Dataset has 697 taxonomies that had results for `Test1`. 67 taxonomies were skipped due to lacking the required files. 347 taxonomies were skipped due to having less than 5 classes. Therefore, 283 taxonomies from the Scior Dataset have been used for the data analysis of this work.

4.2 Data Analysis Environment

The repository containing the data analysis environment created has been provided in the references [16]. The environment created uses Python 3.12. The libraries utilized and their respective use cases have been the following:

- Os: reading files from the Catalog folder and generating/saving new files into their corresponding directories.
- Pandas & numpy: loading data into appropriate data structures.
- Statistics: applying statistical techniques to the collected data.
- Matplotlib: data analysis visualization.

4.3 Strategies Used

The strategies this work uses are an investigation on which meta-properties of the initial seeding will result in more effective enhancement of the provided taxonomy. There are three main categories of strategies to consider while running Scior on an OWL ontology to improve its semantics. These are the *Position*, the *Sortality*, and the *Rigidity* of the initial seeding class used. The strategies that lie within these categories are shown below:

- *Position*: *Root* node (R), *Leaf* node (L), *Intermediate* node (I).
- *Sortality*: *Sortal* (S), *Non-Sortal* (NS).
- *Rigidity*: *Rigid* (RG), *Anti-Rigid* (ARG), *Semi-Rigid* (SRG).

While the *Position* category strategies look to find initial seedings that comply to their *Position* in a given ontology, the *Sortality* and *Rigidity* category strategies look to find initial seedings that have the meta-property aligned with their strategy. The gUFO types that fall under these 5 strategies can be seen in Figure 1 as they are the same in both gUFO and UFO.

5 DATA ANALYSIS RESULTS

The graphs resulting from the data analysis (as shown in Figure 3) are formatted in the same manner. The 8 strategies used all lie on the x-axis. The data resulting from Scior is only analyzed under a

strategy, and present in a strategy's bar, if the initial seeding used for that run falls under the limitations of the said strategy, as it was explained in Subsection 4.3.

The y-axis holds the "Mean of Information Gained Percentage" by applying the strategy to the whole Catalog. It is presented as a percentage as all the data analyzed was chosen to be percentages to be able to generate a more precise description of the effect each strategy had on the taxonomy as a whole since the percentages do not deviate as the size of the taxonomy changes.

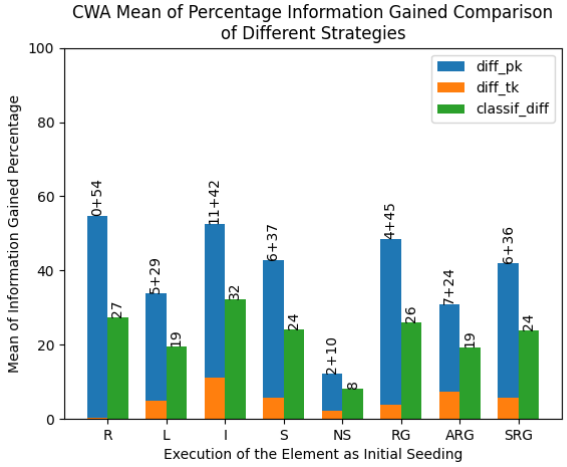

Figure 3: CWA Strategy Comparison

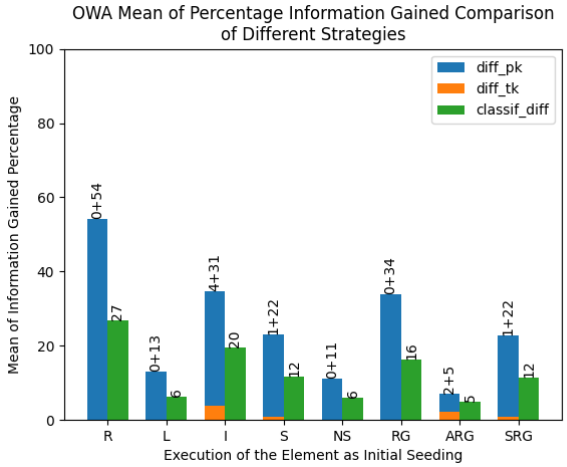

Figure 4: OWA Strategy Comparison

Each graph generates two bars per strategy, as there are two disccusion points when it comes to the effectiveness of a strategy on the initial seeding and consequently the information gained from it.

The first bar is a combination of `diff_pk` and `diff_tk` as it can be seen from a graph's legend. `Diff_pk` refers to the percentage of the classes from the taxonomy that have become partially known, the difference between the percentage of classes partially known before and after running Scior. Partially known means that a class has gained some information regarding its gUFO classification. This is mostly done in two parts, the classifications that the class definitely cannot have, and the classifications the class might have. A partially known class can become a totally known class if more information is present, but it is still better than a totally unknown class, since every bit of information is helpful while classifying a class from the real world.

On the other hand, `diff_tk` refers to the percentage of the classes from the taxonomy that have become totally known. Totally known classes have a single class type they can have and 7 others that they cannot. This is exactly what Scior aims to achieve. Thus, the percentage of totally know classes is an important measure of certainty within a taxonomy.

`Diff_pk` and `diff_tk` percentages have been stacked on top of each other since they count towards the same percentage. The combination of these two will never exceed 100% as it is not possible to have more classes known than there are classes in a taxonomy. They also do not have any intersection, since it is not possible for a class to be both partially and totally known. Therefore, the combination of both statistics allows the graph to reflect the total percentage of classes that Scior created some information about.

The second bar shows the `classif_diff`, which is the percentage of the classifications from the taxonomy that have become known. This can be considered as the percentage of information Scior learned from all the information there was to be learned in the taxonomy. There are 8 class types a class can have (as shown in Figure 1), but there are 14 classifications a class can have (as explained in Section 2).

The classifications extracted from the taxonomy are the most raw form of the information gained from it. Scior then checks if there is a type that fulfills the classifications for a chosen class, giving it the said type if there is one that fulfills them. Thus, it is beneficial to consider the classifications as the amount of information extracted from the ontology while Scior learns about it. Hence, the number of totally known classes presents how fruitful the said information was in identifying the classes of the taxonomy.

5.1 Combined Graph

After inspecting all 8 strategies by their lonesome, it was integral to the data analysis that the combinations of these strategies are also examined. The combinations of the strategies are more potent than a single strategy, as a strategy each from *Position*, *Sortality*, and *Rigidity* can be combined to result in a more specific selection of initial seeding classes. Therefore, allowing the data analysis to pinpoint the information gained from each combination of strategies with higher precision.

Similar to the strategy comparison graphs of both CWA and OWA, the combination graphs also have two bars per strategy. The bars represent the same metrics as the strategy comparison graphs. However, now, there are 18 strategies to examine, as the combination of 3 *Position*, 2 *Sortality*, and 3 *Rigidity* strategies. Thus, the number of strategies lying on the x-axis has been increased to 17 in the CWA graph and 18 in the OWA graph. The CWA graph has one less strategy due to the lack of *Root* elements that are *Sortal* and *Anti-Rigid*. This makes sense since the *Sortal Anti-Rigids*, namely *phases* and *roles*, need to specialize a *kind* and, thus, cannot be *Roots*. However, in an OWA environment, a class can be specializing, be

Tibet Tugay
https://orcid.org/0009-0007-8315-9760

the subclass of, a class that is not yet defined in the taxonomy. Therefore, a *phase* or *role* can in fact be the *Root* of the taxonomy without explicitly specializing a *kind*, resulting in 18 strategies in the OWA environment.

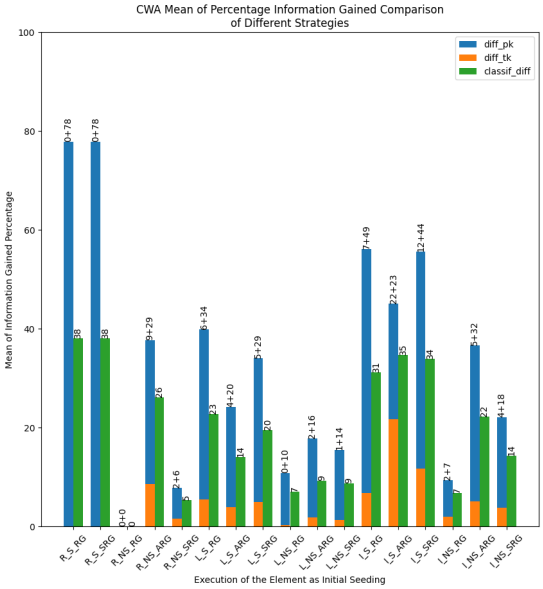

Figure 5: CWA Combination Graph

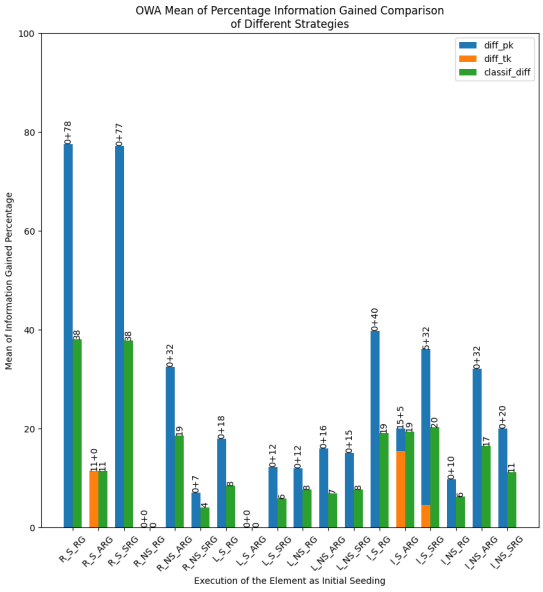

Figure 6: OWA Combination Graph

6 DATA ANALYSIS DISCUSSION

The original paper on Scior [4] has discussed the tool's ability to enhance the semantics of OWL ontologies regarding *correctness* (quality) and *effectiveness* (quantity). The discussion is similar for this work since the results discussed are Scior's results. However, *correctness* is not significant for the sake of the discussion made on this work since it deals with the quality of the inference rules of Scior and that is out of scope for this work.

On the other hand, as the name suggests, *effectiveness* is a great fit for discussing the results of this work. The *effectiveness* of Scior can be measured by "the degree to which Scior can infer, given an initial seeding, the categories of the remaining classes in a taxonomy specified in OWL" [4]. The *effectiveness* of Scior tends to vary between CWA and OWA, as these are the two options for running Scior. Therefore, the discussion considers both cases separately, as it was done in [4].

6.1 Strategy Comparison Discussion

There are two metrics to consider in the discussion of the *effectiveness* of any strategy while running Scior, hence the presence of two bars for each strategy. The percentage of classes known, including both partially and totally known classes, is harder to discuss across a graph, since the comparison between totally known and partially known classes is complex to make.

The main reason for this complexity is the range partially known classes cover. A partially known class can have 1 classification known about it, 13 classifications known about it, or anything in between. Since all the partially known classes get categorized together regardless of how many classifications are known about them, the classifications known will mostly be the benchmark while comparing strategies against each other.

6.1.1 Closed-World Assumption Models. Scior infers all classes and meta-properties defined while running CWA models. Thus, the CWA models tend to have higher results, as the environment has inherently fewer unknowns. This is also demonstrated by the difference between the percentages of Figure 3 and Figure 4.

In CWA models, Scior results in the highest percentage of total information known (`classif_diff`) and totally known classes with a 32% and a 53% respectively, when *Intermediate* strategy is used on the initial seedings. Meanwhile, both the *Root* and *Rigid* strategies also result in similar total information known with 27% and 26% respectively. The *Root* strategy even results in a slightly higher partially known classes with a 54%. On the other hand, *Non-Sortal* strategy is the least effective strategy in CWA models. It has a total information known percentage of 8% and a 12% in classes known. This indicates a clear incentive to avoid using *Non-Sortal* strategy in CWA environments.

6.1.2 Open-World Assumption Models. As mentioned in the previous section, OWA models inherently have more unknowns in their environment. Thus, their percentages are lower than CWA models. As shown in Figure 4, there is a general 12(±2)% drop in total information known. However, the *Root* strategy tends to not deviate from the *effectiveness* of CWA environments, even in an OWA model. This can be seen from the *Root* strategy having the exact percentages for both CWA (Figure 3) and OWA (Figure 4) graphs. This is an intriguing insight into how the *Root* strategy works, as it is considerably higher in both bars than any other strategy in OWA models.

Furthermore, *Intermediate* and *Rigid* strategies are still valid strategies to consider, as they retain 20% and 16% total information known respectively. Lastly, *Leaf* and *Anti-Rigid* strategies have also become strategies to avoid as they join *Non-Sortal* with a 6% and a 5% total information known respectively.

6.2 Strategy Combination Discussion

Similar to the strategy comparison discussion, the effectiveness of the strategy combinations has been examined in two metrics: the percentage of classes known, still including both partially and totally known classes, and the percentage of classifications known, as the benchmark for information gained. Moreover, similar to the strategy comparison graphs, in general, the OWA models yield lower percentages of information gained, due to the higher amount of unknowns the environment enforces on the model.

6.2.1 Closed-World Assumption Models. In CWA models (Figure 5), the highest percentage of information gained has been achieved by both of the *Root-Sortal* (R_S_RG and R_S_SRG) strategies with a 38%. Alongside these, the *Intermediate-Sortal* strategies follow in a close second with classifications known values ranging from 31% to 35%.

However, it is important to note that all of the *Intermediate-Sortal* strategies have resulted in a considerable, 22% in the highest case of I_S_ARG, amount of totally known classes while the *Root-Sortal* strategies both have resulted in 0%. Thus, the selection between *Root* or *Intermediate Positions* depends on whether a higher percentage of classifications or totally known classes are required. The distinction then comes down to whether the most information about the taxonomy in general, or the most complete information about the classes is needed from a taxonomy.

On the other hand, *Sortals* are clearly the better choice when it comes to the *Sortality* of the initial seeding classes. Meanwhile, any choice from *Rigidity* tends to not have a trending effect on the result.

6.2.2 Open-World Assumption Models. In OWA models (Figure 6), the highest percentage of information gained still belongs to the *Root-Sortal* strategies, except R_S_ARG. This is in line with the findings from Figure 4, as the *Root* classes overperformed every other strategy there as well. The 2 *Root-Sortal* classes perform almost equally well in OWA environments compared to CWA environments with only a 1% drop in the classifications known for R_S_SRG.

In contrast, the *Intermediate-Sortal* strategies have dropped considerably in every metric compared to their CWA results, as they did in Figure 4 as well.

In general, *Sortals* tend to have better results than *Non-Sortals*, except for the *Leaf* nodes, where both of them seem to have similar results. Therefore, *Sortals* are clearly the better *Sortality* strategy to choose.

Moreover, *Rigidity* persists in not presenting a distinctly effective option out of *Rigid*, *Anti-Rigid*, and *Semi-Rigid*. There is an opportunity to refer to *Anti-Rigids* as the superior choice due to the higher percentages in totally known classes and classifications known. However, the data from the *Leaf* nodes and the *Root-Sortal* classes dispatch this notion as the *Anti-Rigids* are the worst performers in the respective categories. This suggests that an opportunity to analyze this behavior with a larger dataset might be worth investigating.

7 CONCLUSION

This work aimed to realize the most effective strategies for running Scior to enhance the semantics of OWL ontologies. This was especially important to ensure a higher standard for interoperability and reusability of OWL ontologies. To do so, the Scior results have been analyzed in relation to the initial seeding used to provide the said results. Throughout the data analysis, 8 strategies related to the *Position*, *Sortality*, and *Rigidity* of the initial seeding have been examined.

The results and the subsequent discussion of them suggest that there are indeed better strategies to adopt or prioritize while enhancing the semantics of OWL ontologies with Scior. There is a slight variation for these better strategies in the cases of CWA and OWA models since these two have a clear difference in their environment and how they approach classifying a class.

Nevertheless, *Root*, *Intermediate*, *Sortal*, and *Rigid* strategies return better results than the other strategies in general, even though the *Rigid* strategy presents no clear effect when it is combined with other strategies. The key takeaway for CWA models is to avoid using the *Non-Sortal* strategy, as it underperforms heavily compared to the other strategies. On the other hand, for OWA, one should definitely prioritize *Root* strategy, as it is clearly overperforms all other strategies in this environment.

7.1 Future Work

The future works for this work mostly arise from the limitations it suffered from. The most significant of which is to apply the data analysis to the updated version of the Catalog. Scior Dataset was created more than a year ago and during that time the Catalog has evolved to hold more models than before. The additional models would add to the knowledge generated from this data analysis. If this was done by running Scior Tester on the new Catalog, the files that are missing would also be generated, allowing the data analysis to not skip some of the taxonomies due to lacking the required files. This would give a broader view of the Catalog in the data analysis, possibly resulting in new insights.

The other limitation this work had was skipping taxonomies that had less than 5 classes. This was done to preserve the percentage from being diverted too much, as the low number of classes skews the percentages. This could be dealt with by either using the number of classes and classifications instead of percentages or adding those statistics to the current data analysis. Nonetheless, the data resulting from that might be less reliable as the raw number of classes/classifications would then be skewed by taxonomies with a higher number of classes.

Another topic to explore is the patterns that can be extracted from the prioritization of the inference rules Scior uses. This would be more of a technical topic to analyze as it requires knowledge of how Scior operates internally. However, it might have significant results revolving around which inference rules are better to prioritize. This future work could be combined with what this work

offers to find the most optimal strategy to enhance OWL semantics with by running Scior.

Furthermore, an investigation into other properties of the initial seeding classes can be used to improve the strategies discussed in this work. An example of this is the number of subclasses and superclasses a class has. After applying a combination of *Position*, *Sortality*, and *Rigidity* strategies, choosing the class with the maximum or minimum number of subclasses or superclasses might make the strategies even more effective. This work attempted to explore this aspect of the initial seeding class as well, however, it was deemed out of scope in the end.

Lastly, there is an argument to be made for training a small pattern detection algorithm to analyze and produce patterns that can be used to optimize the initial seeding even more. However, this was outside the scope of this work due to the time constraints encompassing this project.

REFERENCES

[1] Syed Ali, Dominik Bork, and Giancarlo Guizzardi. 2023. Enabling Representation Learning in Ontology-Driven Conceptual Modeling using Graph Neural Networks.

[2] João Paulo A. Almeida, Ricardo A. Falbo, Giancarlo Guizzardi, and Tiago P. Sales. 2020. *gUFO: A Lightweight Implementation of the Unified Foundational Ontology (UFO).* Technical Report. online: https://purl.org/nemo/doc/gufo. Accessed: 2024-04-25.

[3] Pedro Barcelos, Tiago Prince Sales, Mattia Fumagalli, Claudenir Fonseca, Isadora Valle Sousa, Elena Romanenko, Joshua Kritz, and Giancarlo Guizzardi. 2022. *A FAIR Model Catalog for Ontology-Driven Conceptual Modeling Research.* 3–17. https://doi.org/10.1007/978-3-031-17995-2_1

[4] Pedro Barcelos, Tiago Prince Sales, Elena Romanenko, João Almeida, Gal Engelberg, Dan Klein, and Giancarlo Guizzardi. 2023. Inferring Ontological Categories of OWL Classes Using Foundational Rules.

[5] Francisco García-Peñalvo, Juan Garcia, and Roberto Therón. 2011. Analysis of the OWL ontologies: A survey. *Scientific research and essays* 6 (09 2011), 4318–4329.

[6] Giancarlo Guizzardi. 2019. Ontology, Ontologies and the "I" of FAIR. *Data Intelligence* 2 (11 2019), 181–191. https://doi.org/10.1162/dint_a_00040

[7] Giancarlo Guizzardi, Alessander Benevides, Claudenir Fonseca, Daniele Porello, João Almeida, and Tiago Prince Sales. 2022. UFO: Unified Foundational Ontology. *Applied Ontology* (01 2022). https://doi.org/10.3233/AO-210256

[8] Annika Jacobsen, Ricardo de Miranda Azevedo, Nick Juty, Dominique Batista, Simon Coles, Ronald Cornet, Mélanie Courtot, Mercè Crosas, Michel Dumontier, Chris T. Evelo, Carole Goble, Giancarlo Guizzardi, Karsten Kryger Hansen, Ali Hasnain, Kristina Hettne, Jaap Heringa, Rob W.W. Hooft, Melanie Imming, Keith G. Jeffery, Rajaram Kaliyaperumal, Martijn G. Kersloot, Christine R. Kirkpatrick, Tobias Kuhn, Ignasi Labastida, Barbara Magagna, Peter McQuilton, Natalie Meyers, Annalisa Montesanti, Mirjam van Reisen, Philippe Rocca-Serra, Robert Pergl, Susanna-Assunta Sansone, Luiz Olavo Bonino da Silva Santos, Juliane Schneider, George Strawn, Mark Thompson, Andra Waagmeester, Tobias Weigel, Mark D. Wilkinson, Egon L. Willighagen, Peter Wittenburg, Marco Roos, Barend Mons, and Erik Schultes. 2020. FAIR Principles: Interpretations and Implementation Considerations. *Data Intelligence* 2, 1-2 (01 2020), 10–29. https://doi.org/10.1162/dint_r_00024 arXiv:https://direct.mit.edu/dint/article-pdf/2/1-2/10/1893430/dint_r_00024.pdf

[9] Cartik R. Kothari and David J. Russomanno. 2008. Enhancing Owl Ontologies With Relation Semantics. *International Journal of Software Engineering and Knowledge Engineering* 18, 03 (2008), 327–356. https://doi.org/10.1142/s0218194008003660 arXiv:https://doi.org/10.1142/S0218194008003660

[10] Deborah L McGuinness, Frank Van Harmelen, et al. 2004. OWL web ontology language overview. *W3C recommendation* 10, 10 (2004), 2004.

[11] Georgios Meditskos and Nick Bassiliades. 2008. Combining a DL Reasoner and a Rule Engine for Improving Entailment-Based OWL Reasoning. In *The Semantic Web - ISWC 2008*, Amit Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy Finin, and Krishnaprasad Thirunarayan (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 277–292.

[12] Jing Mei, Elena Paslaru Bontas, and Zuoquan Lin. 2005. OWL2Jess: A Transformational Implementation of the OWL Semantics. In *Parallel and Distributed Processing and Applications - ISPA 2005 Workshops*, Guihai Chen, Yi Pan, Minyi Guo, and Jian Lu (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 599–608.

[13] Pedro Paulo F. Barcelos, Tiago Prince Sales, Elena Romanenko, João Paulo A. Almeida, Gal Engelberg, and Dan Klein. [n. d.]. *Scior Dataset.* https://github.com/unibz-core/Scior-Dataset

[14] Pedro Paulo F. Barcelos, Tiago Prince Sales, Elena Romanenko, João Paulo A. Almeida, Gal Engelberg, and Dan Klein. [n. d.]. *Scior Tester.* https://github.com/unibz-core/Scior-Tester

[15] Goutam Kumar Saha. 2007. Web ontology language (OWL) and semantic web. *Ubiquity* 2007, September, Article 1 (sep 2007), 1 pages. https://doi.org/10.1145/1295289.1295290

[16] Tibet Tugay. [n. d.]. *Scior Data Analysis Environment.* https://purl.org/ttugay-scior-data-analysis