An Efficient Approach for Semantic Relatedness Evaluation based on Semantic Neighborhood

1st Alcides Lopes
Institute of Informatics
UFRGS
Porto Alegre, Brazil
agljunior@inf.ufrgs.br

2nd Renata Alvarenga
Institute of Geosciences
UFRGS
Porto Alegre, Brazil
rsakuchle@inf.ufrgs.br

3rd Joel Carbonera
Institute of Informatics
UFRGS
Porto Alegre, Brazil
jlcarbonera@inf.ufrgs.br

4th Mara Abel
Institute of Informatics
UFRGS
Porto Alegre, Brazil
marabel@inf.ufrgs.br

***Abstract*—In the context of natural language processing and information retrieval, ontologies can improve the results of the word sense disambiguation (WSD) techniques. By making explicit the semantics of the term, ontology-based semantic measures play a crucial role to determine how different ontology classes have a similar meaning. In this context, it is common to use semantic similarity as a basis for WSD. However, the measures generally consider only taxonomic relationships, which affects negatively the discrimination of two ontology classes that are related by the other relationship types. On the other hand, semantic relatedness measures consider diverse types of relationships to determine how much two classes on the ontology are related. However, these measures, especially the path-based approaches, has as the main drawback a high computational complexity to be calculated in query execution time. Also, for both types of semantic measures, it is unpractical to store all similarity or relatedness values between all ontology classes in memory, especially for large ontologies. In this work, we propose a novel approach based on semantic neighbors that aim to improve the query time in path-based semantic measures without losing their effectiveness in relatedness analysis. We also propose an efficient algorithm to calculate the semantic distance between two ontology classes. We evaluate our proposal in WSD using a pre-existent domain ontology for well-core description. This ontology contains 929 classes related to rock facies and a set of sentences from four different corpora about geology in the Oil&Gas domain. In the experiments, we compared our approach with state-of-the-art semantic relatedness measures, such as path-based, feature-based, information content, and hybrid methods regarding the F-score, query time and the total number of classes in memory. The experimental results show that the proposed method obtains F-score gains in WSD, as well as an improvement in the query time concerning the traditional path-based approaches. Also, we reduce the total number of classes stored in memory for each ontology class.**

***Index Terms*—Semantic relatedness, semantic neighbors, word sense disambiguation**

I. Introduction

Word sense disambiguation (WSD) is the task of automatically identifying the intended sense of an ambiguous term based on the context in which the term is used [1]. In this context, two broad categories of semantic measures try to disambiguate a term based on the ontology structure [2]: semantic similarity, in which generally explore only taxonomic relationships; and semantic relatedness, which explores other types of relationships than taxonomic ones.

The semantic similarity approaches have as the main drawback the inability to discriminate two ontology classes in situations where most of their relationships are not of the taxonomic type. On the other hand, the semantic relatedness measures do not present the disadvantage of similarity measures, but these measures, especially the path-based approaches, have high computational complexity in query time. In this context, we refer to query time as the time to search, calculate, and return the value of similarity and relatedness between two ontology classes. Also, for both types of semantic measures, it is unpractical to store all values of similarity or relatedness between all ontology classes in memory, especially for large ontologies.

We emphasize the semantic measures based on paths because we believe that they are the best choice for relatedness evaluation. In the path-based measures, it is possible to express how much related two ontology classes are, according to the semantic distance between them in the ontology.

In this work, we propose a novel approach that aims to improve the query time in path-based semantic measures without losing their effectiveness in relatedness analysis. To achieve this improvement, we describe an algorithm to calculate the semantic distance between two ontology classes by adapting the semantic path patterns proposed in [3]. We use the direction of these semantic path patterns in two particular situations. The first one is before a query, where we use a set of path patterns to discover the semantic neighbors of a given ontology class (source class). The semantic neighbors of a given class c are a set classes N, such that, each $n_i \in N$ is directly related to c through direct semantic paths (as described in Section III-B). To improve the query time and the memory consumption, we store only the relationships that represent the direct semantic paths between each ontology class and its respective semantic neighbors. Already on the second occasion, we used a different set of semantic path patterns to calculate the semantic distance through indirect semantic paths (as described in Section III-C).

In order to evaluate our proposal, we use the method described in [2] to perform the word sense disambiguation (WSD) of the terms that name different ontology classes. We extract the context window where these terms occur from four different corpora in the Oil&Gas domain using a domain

2375-0197/19/$31.00 ©2019 IEEE
DOI 10.1109/ICTAI.2019.00052

ontology for core description defined by Lorenzatti *et al.* [4] to support the Strataledge®[1]. In our experiments, we evaluated the results of different ontology-based semantic measures, such as path-based, feature-based, information content, and hybrid approaches, according to the F-score obtained in the WSD task. Also, we compare the query time and the memory consumption of path-based approaches with and without our proposal. As a result, we obtain a considerable reduction in the total number of classes in memory, as well as we get a significant improvement in the query time compared to traditional path-based approaches and often with better F-score result in WSD.

We organize this paper as follows. Section II briefly reviews the state-of-the-art approaches for computing the semantic similarity and relatedness and their main problems in the relatedness evaluation. In Section III, we describe our approach to find the semantic neighbors of an ontology class and how we use them in our semantic distance measure. In Section IV-B, we show our experiments in WSD and their results. Finally, Section V summarizes our conclusions about the proposed approach.

II. Related Works

In the last years, many researchers have proposed ontology-based semantic measures, especially with the increasing interest in the ontology-based applications [5]. Ontologies raise the interest of the semantic similarity and relatedness research community as they offer a structured and unambiguous representation of the knowledge. In this work, we classified the ontology-based semantic measures into four groups: path-based, information content (IC), feature-based, and hybrid approaches. Also, we present a brief discussion of the drawbacks of these approaches in relatedness evaluation.

The path-based approaches are based on the distance or length of the shortest path between two ontology classes to evaluate its similarity or relatedness [6]. This length is the sum of the weight associated with the edges (relationships) which compose the path. When the edges are not weighted, the length of the shortest path is the number of edges it contains. In addition to the length of the shortest path, researchers apply the relative depth of the classes [7] [8] or the local density of the class in ontology [9] for similarity or relatedness calculation. These approaches have as the main disadvantage the complexity in evaluating all paths in query time. In [3], the authors classify the ontology relationships in upward (UR), downward (DR), and horizontal (HR). Also, in relatedness evaluation, these authors consider a set of correct path patterns through the sequence of relationships UR, UR-DR, UR-HR, UR-HR-DR, DR, HR-DR, and H. The notation "-" means the direction of a path, for example, in UR-DR paths a sequence of DR relationships follows a sequence of UR relationships.

The information content (IC) approaches describe how specific and informative a concept is and the semantic similarity between two ontology classes depends on the amount of information two concepts have in common. The most specific common abstraction gives this shared information, i.e., the super-class that subsumes both classes with the highest IC value [10]. There are two main models for this approach: the corpus-based model and the intrinsic model. Resnik's [10] proposed the former, in which the frequency of an ontology class in a given corpus is its IC value. The main disadvantage of this approach is the dependence of the existence of corpora and the probabilistic influence of the term frequencies in corpora. The later model was proposed to avoid the disadvantages of the former by using the ontology structure, where the IC value increase regarding the relative depth of a class in ontology [11] [9] [12] [13]. However, in general, the main limitation of this approach is that it uses only taxonomic relationships for defining the similarity. The latter approach may fail to evaluate the relatedness value in ontologies, where there are not many related classes through taxonomic relationships.

The main idea underlying the feature-based approaches is that similarity increases as more features in common two classes share [14]. In the state-of-the-art, usually, the authors consider as the features of an ontology class, the set of classes related to this class. Basically, the feature-based measures perform an inference process for each class relationship in order to find a more concise set of features. However, there are no general rule about how combine different relationship types during the inference process. In another context, some works propose modified versions of the Tversky's model to avoid the dependence on smoothing factors [15]. Other works propose the relatedness evaluation based on the overlap of the textual information present in the descriptions (glosses) of the classes [16]. However, this method is sensitive to variations of the textual description of the classes.

Hybrid approaches take advantage of the combination of the previously mentioned methods. For example, in [11] a path-based measure and an IC measure are integrated to evaluate the semantic relatedness between two ontology classes. However, the hybrid approaches also inherit the drawbacks of the measures that are combined.

It is important to notice that there is a significant limitation in the application of IC-based measures in relatedness evaluation because they usually exploit only taxonomic features of the ontology. The feature-based approaches, on the other hand, are very sensitive to how feature sets are selected and used. Lastly, in path-based approaches, the computation of the shortest path considering all the ontology relationships is a task with a high computational cost in query time. The drawback of storing all relatedness values between all ontology classes in memory is a common problem in any application that depends on a relatedness measure [17]. In order to overcome these challenges, in this work, we use the directions of the path patterns proposed in [3] for finding and storing only the semantic neighbors of an ontology class in memory. Also, we propose an approach to calculate the semantic distance between two ontology classes considering the restrictions of the path patterns in order to improve the query time of the traditional path-based approaches. In the next section, we

[1]Strataledge is a trademark of Endeeper Co. www.strataledge.com

present our proposal in detail.

III. A Semantic Neighborhood Approach to Semantic Relatedness Evaluation

In this section, we present our proposal. In the first subsection, we present the background notions adopted in our approach. In the second subsection, we describe how we obtain the semantic neighbors of an ontology class. In the third and last subsection, after we have all the semantic neighbors of the ontology classes in memory, we describe our approach to determine the semantic distance between two ontology classes.

A. Notations

In this section, we present the literature notions about the semantic graphs and the paths that exist in the semantic graphs through their edges.

Definition 1 (Semantic Graph). *Let $G = (V, E)$ be a directed graph that represents an ontology O where V is a finite set of vertexes that represents the ontology classes, and E is a finite set of edges that represent the ontology relationships. In an ontology that has only binary relationships, the tuple (c_i, r, c_j) describes an edge e, where $c_i \in V$ and $c_j \in V$ and c_i is the subject (or the source vertex), r is the predicate (or the relation) and c_j is the object (or the target vertex).*

Definition 2 (Path). *Let $G = (V, E)$ be a directed graph. A path $P(c_i, c_j)$ between $c_i, c_j \in V$ is a sequence of edges $\{e_1, ..., c_k\} \in E$ with size n that relates the vertexes c_i and c_j such that there is no repetition of visited vertexes in the sequence.*

B. Semantic Neighbors

In this section, we describe our view about the types of relationships presented in an ontology and the types of direct semantic paths built through the relationships composition. Also, we present a set of assumptions to support our proposal to find the semantic neighbors of an ontology class.

Assumption 1 (Inverse Relationship). Since the edge e is oriented, we denote r^- the type of relation that has the inverse semantic of r and e^- the inverse semantic edge of the direct edge e. We consider that any relationship (c_i, r, c_j) implicitly implies (c_j, r^-, c_i). For example, the hierarchical relationship *(sedimentary rock, sub-class of, rock)* implies the inverse hierarchical relationship *(rock, super-class of, sedimentary rock)*, considering *sub-class of*$^- =$ *super-class of*. The same situation occurs when considering a sequence of edges. For example, the part-whole relationships *(cerebellum, part of, brain)* and *(brain, part of, person)* implies the inverse part-whole relationships *(brain, has part, cerebellum)* and *(person, has part, brain)*. The inverse relationship has not necessarily the same logical properties as the direct relation. The resulting graph G is strongly connected, i.e., any vertex c_i is reachable from any other vertex c_j, and vice versa.

Assumption 2 (Distinct Relationships). Two ontological relationships are distinct if they have different logical properties or if the relations have different names. We are aware that certain relationships show different names but have the same semantics (synonymous relationships), but it is hard to distinguish these relationships in these cases. Besides, we classify the relationships of an ontology into four types:

- **Equivalent Relationships (ER):** any relationship that has logical properties of reflexivity, symmetry, and transitivity and conveys the idea that one class c_i is semantically equivalent to another class c_j. This type of relationship must have the greatest relatedness value as possible between two ontology classes;
- **Hierarchical Relationships:** this type of relationship conveys the idea of hierarchy among the related classes (e.g., is-a, sub-class of). We subdivide this type of relationship into Upward and Downward. Upward relationships **(UR)** start from a more specific vertex to a more generic vertex (e.g., a relationship through a *sub-class of* relation). Downward relationships **(DR)** start from a more generic vertex to a more specific vertex (e.g., a relationship through a *super-class of* relation);
- **Horizontal Relationships (HR):** is classified in this category, any relationship that is not possible to classify in the previously described types, such as part-whole, characterization, and constitution relationships. We are aware that this category includes very different semantic relations. In future works, we aim to split this category for dealing with different classes of relationships in more specific ways.

Assumption 3 (Direct Semantic Path). We assume that there is a direct semantic path between two classes c_i e c_j when the relationships previously described or a combination of them directly relate two classes c_i and c_j. In this work, we consider that the set of the possible direct semantic paths, in the search for the semantic neighbors, are through the following relationships: ER, UR, DR, HR, UR-HR, HR-DR, and UR-HR-DR. Algorithm 1 is used to obtain the set of relationships that describe this direct semantic path between a given ontology class and all its semantic neighbors. In the next assumption, we detail how to get the direct semantic paths and how they influence the relatedness value between two ontology classes.

Assumption 4 (Local Distance). The local distance (LD) between two ontology classes c_i and c_j, in a direct semantic path, is equal to the length of the shortest path between them. This value expresses how related these two classes are. In this work, we convert a direct semantic path in a relationship that has as property the local distance. This property has the same value as the local distance of the converted direct semantic

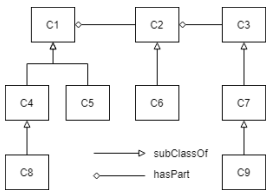

Fig. 1. An example of the classes and their relationships.

path. However, the local distance value is dependent on the type of the direct semantic path analyzed:

- **ER path:** occurs only through a sequence of ER-type relationships. In Algorithm 1, the function $getER$ returns the set o relationships between a class e and all its semantic neighbors related through ER path. As discussed in Assumption 2, the relatedness value between two classes that are related by this path type always has the greatest possible relatedness value, and then its LD value is equal to 0;
- **UR path:** occurs only through a sequence of UR-type relationships. This type of direct semantic path implies that whenever we refer to the source class, we are implicitly referring directly to the target class. Thus, the LD value for this type of direct semantic path is always greater than the LD value of ER path and lower or equal to the minimum LD value of the DR path. In Algorithm 1, the function $getUR$ returns the set of relationships between a class e and all its semantic neighbors related through UR path;
- **DR path:** occurs only through a sequence of DR-type relationships. Unlike the UR semantic path, we can not use the same logic for this semantic path. For example, in Fig. 1, the *C1* class is intuitively more related to the classes *C4* and *C5* than to the *C8* class or any class that subsumes the *C4* or *C5* classes. Thus, the LD value for this path type is equal to the number of edges in the DR path. In Algorithm 1, the function $getDR$ returns the set o relationships between a class e and all its semantic neighbors related through the DR path;
- **HR path:** occurs only through a sequence of the HR-type relationships. For example, in Fig. 1, considering that the *C1* represents the class *Person*, the *C2* represents the class *Brain*, and *C3* represents the class *Cerebellum*, the *C3* is intuitively more related to the *C2* than to *C1*, as well as the *C1* is more related to the *C2* than to *C3*. Thus, we consider the LD value is equal to the length of the HR path. In Algorithm 1, the function $getHR$ returns the set o relationships between a class e and all its semantic neighbors related through the HR path;
- **HR-DR path:** this composition of relationships implies that a sequence of DR-type relationships follows a sequence of HR-type relationships. For example, in Fig. 1, consider that the *C2* represents the class *Heart*, the *C1* represents the class *Animal*, and the relation *part of* is inverse of the *has part* relation, intuitively the *Heart* is as closely related to the *Animal*, as well as to all classes that subsume the *Animal* (e.g., Wolf, Mammalian Animals, among others). Thus, we consider the LD value is equal to the length of the HR path between the evaluated classes;
- **UR-HR path:** this composition of relationships implies that a sequence of UR-type relationships is required, followed by a sequence of HR-type relationships. For example, in Fig. 1, consider that the *C2* represents the class *Heart*, and the *C1* represents the class *Animal*, intuitively the *Animal* is as closely related to the *Heart* class, as well as to all classes that subsume the *Animal* class (e.g., Wolf, Mammalian Animals, among others). Thus, we consider the LD value is equal to the length of the HR path between the evaluated classes;
- **UR-HR-DR path:** this composition of relationships implies that a sequence of UR-type relationships is required, followed by a sequence of HR-type relationships, and then followed by a sequence of DR-type relationships. Since we consider LD values of HR-DR and UR-HR paths are equals to the length of the HR path, then the same logic is applied to this type of direct semantic path, i.e., we consider the LD value of UR-HR-DR path is equal to the length of the HR path between the evaluated classes.

Based on the descriptions of the assumptions 3 and 4, Algorithm 1 perform the search for the set of relationships SN between an ontology class c_i and all its semantic neighbors, where: in line 3 the ER-type relationships are added in SN; in line 6 the DR-type relationships are added in SN; in line 9 the HR-type relationships are added in SN; in line 16 the UR-type relationships are added in SN; and in the line 19 the UR-HR-type relationships are added in SN. However, since relationships types HR-DR and UR-HR-DR use the LD value of their respective HR relationship, then, in the lines 12 and 22 of Algorithm 1, the relationships HR are added with the target class of DR relationship in SN.

It is important to note that the sequence of relationships covered by the functions $getER$, $getDR$, $getUR$, and $getHR$ of Algorithm 1 should not be distinct (as described in Assumption 2).

Algorithm 1 finds all the relationships between every class on ontology and all its semantic neighbors, keeping them in memory. Since, for each class, only the relationships between it and its neighbors are stored, this decrease the amount of memory needed and decreases the query time. We do this step before any query of relatedness evaluation is performed. However, since a given class may not contain the local distance values with all ontology classes through a direct semantic path, then it is necessary to find this value through an indirect semantic path. We perform this step in query time, as presented in the next subsection.

C. Semantic Distance

In order to solve the problem of the evaluation of two ontology classes that are not neighbors, we propose the use of the nearest common neighbor (NCN) as an intermediary class. Also, in Section III-B, we discover the semantic paths through the path patterns ER, UR, DR, HR, UR-HR, HR-DR, and UR-HR-DR relationships. Thus, by evaluating the indirect paths (described below), the UR-DR path pattern is also evaluated.

Assumption 4 (Common Neighbors). Consider that the function SN of Algorithm 2 represents the output of Algorithm 1, sn_i the set of semantic neighbors of the class c_i, excluding the DR-type relationships, and sn_j the set of semantic neighbors of the class c_j, excluding the UR-type

Input: class c_i
Output: The set of relationships SN from the class c_i to its semantic neighbors

```
1  SN ← ∅
2  for ER ∈ getER(c_i) do
3  |  add ER to SN
4  end
5  for DR ∈ getDR(c_i) do
6  |  add DR to SN
7  end
8  for HR ∈ getHR(c_i) do
9  |  add HR to SN
10 |  c_j ← HR.target
11 |  for DR ∈ getDR(c_j) do
12 |  |  add HR to SN with DR.target
13 |  end
14 end
15 for UR ∈ getUR(c_i) do
16 |  add UR to SN
17 |  c_j ← UR.target
18 |  for HR ∈ getHR(c_j) do
19 |  |  add HR to SN
20 |  |  c_k ← HR.target
21 |  |  for DR ∈ getDR(c_k) do
22 |  |  |  add HR to SN with DR.target
23 |  |  end
24 |  end
25 end
26 return SN;
```

Algorithm 1: Algorithm applied to find the neighbors of a class c_i through the direct semantic paths.

relationships, then the function $CN(sn_i, sn_j)$ (line 11 of Algorithm 2) returns the set of common neighbors between sn_i and sn_j. We exclude some types of relationship from sn_i and sn_j to maintain the restrictions of the path patterns described in Section II and III-B.

Assumption 5 (Nearest Common Neighbor). We assume that the nearest common neighbor is the class whose value $distance$ is the smallest, within the set of common neighbors $ce \in CNs$ between sn_i and sn_j. The sets sn_i and sn_j are the neighbors of the classes c_i and c_2, respectively, and $distance = LD(cc2c_i) + LD(cc2c_j)$ (line 15 of Algorithm 2), where $cc2c_i$ is the relationship between cc and c_i and $cc2c_j$ is the relationship between cc and c_j (lines 13 and 14 of Algorithm 2, respectively).

Assumption 6 (Indirect Semantic Path). We assume that two classes c_i and c_j do not have a direct semantic path between them but are related to the same common neighbor cc class, and this common neighbor is the nearest common neighbor of both. In this case, the local distance (LD) between c_i and c_j is equal to the sum of the LD value of the relationship between cc and c_i and the LD value of the relationship between cc and c_j.

Based on all the assumptions described in the sections III-B and III-C, we propose Algorithm 2 to compute the semantic distance between two ontology classes in query time. In this algorithm, the semantic distance between two classes c_i and c_j assumes one value of the three possible situations:

- In the first situation, the two analyzed classes are the same (lines 2-4 of Algorithm 2). Thus, the semantic distance between them is equal to 0;
- In the second situation, as discussed in Section III-B, there is a direct semantic path between the two analyzed classes (lines 5-9 of Algorithm 2). Thus, we retrieve, from memory, the semantic neighbors sn_i of the class c_i. If the class c_j is present in the relationships between c_i and its semantic neighbors sn_i, then the semantic distance is equal to the LD value of the relationship between c_i and c_j;
- In the third occasion, as discussed in this section, an indirect semantic path exists between the two analyzed classes (lines 5 and 10-20 of Algorithm 2). Thus, we retrieve, from memory, the semantic neighbors sn_j of the class c_j and we get the common neighbors between sn_i and sn_j. From the set of common neighbors, we search the nearest common neighbor cc between c_i and c_j. Finally, the semantic distance is equal to the sum of the LD value of the relationship between cc and c_i and the LD value of the relationship between cc and c_j.

The main differentials of our semantic distance approach are that we reduce the number of possible paths used, and we ensure the relationships between all ontology classes through a direct or indirect semantic path.

Input: Source class c_i and Target class c_j
Output: Length value $minDistance$

```
1  minDistance ← MAXVALUE
2  if c_i ≡ c_j then
3  |  minDistance = 0
4  else
5  |  sn_i ← SN(c_i)
6  |  if c_j ∈ sn_i then
7  |  |  c_i2c_j ← c_i.getRelationshipTo(c_j)
8  |  |  minDistance = LD(c_i2c_j)
9  |  else
10 |  |  sn_j ← SN(c_j)
11 |  |  CNs ← CN(sn_i, sn_j)
12 |  |  for cc ∈ CNs do
13 |  |  |  cc2c_i ← cc.getRelationshipTo(c_i)
14 |  |  |  cc2c_j ← cc.getRelationshipTo(c_j)
15 |  |  |  distance ← LD(cc2c_i) + LD(cc2c_j)
16 |  |  |  if distance < minDistance then
17 |  |  |  |  minDistance ← distance
18 |  |  |  end
19 |  |  end
20 |  end
21 end
22 return minDistance;
```

Algorithm 2: Semantic Distance Algorithm

IV. Experiments and Analysis

In this section, we evaluate our proposal for word sense disambiguation (WSD). In [2], the authors propose the use of the ontology structure to disambiguate a term that describes two or more ontology classes (polysemic term) using a *context window*. The *context window* consists of the target ontology classes of the disambiguation and some number (window size) of ontology classes that occur to the left and right in a given sentence. In the experiments, we adopted the domain ontology of Strataledge®. This ontology has 929 concepts that support the detailed and systematic description of sedimentary facies in drill cores. It includes all classes of lithologies, textures, structures, and fossil content defined according to a top-level ontology. In this ontology, there are 358 occurrences of polysemic terms.

In the next sections, we described the datasets from which we extracted the sentences that occur the polysemic terms of the Strateledge® ontology. Also, we evaluate our proposal according to the F-score results in WSD. In addition, we evaluate the memory consumption to store the ontology classes, as well as the query time of our semantic distance measure (described in Section III-C).

A. Evaluated datasets

In this work, we extract the sentences that contain the polysemic terms of the Strateledge® ontology from four different corpora:

- **D1 (Polvo Project):** the Polvo Project is a Geological study developed by the Geoscience Institute of Federal University of Rio Grande do Sul in cooperation with Maersk Energia Company. The project comprises an integrated study of Petrology, Sedimentology, Seismic Sequence Stratigraphy and Biostratigraphy, developed with data from the Polvo and Peregrino fields area, Campos Basin, Brazil. In this corpora, we use only the final report document;
- **D2 (Scherer scientific articles):** this repository is a set of papers written by one of the stratigraphers that participated in the creation of the domain ontology of Strataledge®. The articles describe the analysis of facies architecture and the sequence stratigraphy of some fluvial and eolian reservoirs;
- **D3 (Sedimentary Geology journal):** this journal covers all aspects of sediments and sedimentary rocks at all spatial and temporal scales. The collection of articles must make a significant contribution to the field of study and must place the research in a broad context so that it is of interest to the diverse, international readership of the journal. This dataset includes four papers of the Sedimentary Geology journal (volume 379);
- **D4 (Sedimentology journal):** this journal publishes ground-breaking research from across the spectrum of sedimentology, sedimentary geology, and sedimentary geochemistry. This dataset includes the papers of the Sedimentology journal (volume 66, issue 4).

During the sentences extraction, in order to facilitate the string matching, we performed the stop-word removal and stemming on the text of the input document and the polysemic term. After having all the extracted sentences, a geologist classified the sentences if they are according to the context of the Strataledge® ontology. We do that because, in this paper, we aim to solve the problem of polysemy and not of the homonymy. Finally, are extracted a total of 1732 sentences where 920 are according to the domain of the Strataledge® ontology and are considered to perform the WSD (Table I).

TABLE I
THE CHARACTERISTICS OF EACH EVALUATED DATASET.

	D1	**D2**	**D3**	**D4**
Total no. of extracted sentences	325	433	288	686
No. of considered sentences	109	341	96	374

B. Experimental results and analysis

We conducted three categories of experiments: the first contains the comparative experiment using the ontology-based semantic measures described in the state-of-the-art. We perform the word sense disambiguation (WSD) with these measures, using the WSD algorithm proposed in [2] and the set of sentences described in Section IV-A; the second experiment is a comparative experiment of different approaches illustrating the average number of classes required in memory for each ontology class; the third experiment is a comparative experiment of different approaches focused on evaluating the average query time of a relatedness evaluation between all Strataledge® ontology classes.

TABLE II
THE ONTOLOGY-BASED SEMANTIC MEASURES EVALUATED IN THE EXPERIMENTS

ID	**Approach**	**Type**	**Parameters**
(1)	Resnik-Seco [10], [12]	IC(intrinsic)	-
(2)	Resnik-Zhou [10], [9]	IC(intrinsic)	$k = 0.5$
(3)	Resnik-Pirro [10], [13]	IC(intrinsic)	$\alpha = 0.5$, and $\beta = 0.5$
(4)	Tversky [14]	Feature	$\alpha = 0.5$, and $\beta = 0.5$
(5)	Likavec [15]	Feature	-
(6)	Rada [6]	Path	-
(7)	W&P [18]	Path	-
(8)	Li [7]	Path	$\alpha = 0.5$, and $\beta = 0.5$
(9)	Liu [8] Strat 1	Path	$\alpha = 0.5$, and $\beta = 0.5$
(10)	Liu [8] Strat 2	Path	$\alpha = 0.5$, and $\beta = 0.5$
(11)	Hao [19]	Path	$\alpha = 0.5$, and $\beta = 0.5$
(12)	Cai [11] Strat 1	Hybrid	$\alpha = 0.5$, $\beta = 1$, and $\lambda = 0.5$

There are 12 approaches covered in the first experiment (Table II) divided into two sets. In the first set, are evaluated the original approaches as described in the state-of-the-art. Already in the second set, are evaluated the adaptation of the original path and feature-based measures with our proposal, totalizing 21 different semantic measures. We perform the adaptation of feature-based measures taking into account the

TABLE III
F-SCORE OF WSD IN % OF THE EVALUATED SEMANTIC MEASURES WITH AND WITHOUT OUR APPROACH.

Semantic Measure	Window Size															
	2				4				6				8			
	D1	D2	D3	D4	D1	D2	D3	D4	D1	D2	D3	D4	D1	D2	D3	D4
The original proposal:																
(1)	0.18	0.24	0.22	0.27	0.14	0.31	0.24	0.35	0.24	0.35	0.32	0.38	0.26	0.40	0.30	0.42
(2)	0.21	0.24	0.25	0.28	0.24	0.34	0.24	0.36	0.34	0.41	0.34	0.41	0.34	0.45	0.34	0.45
(3)	0.21	0.25	0.25	0.29	0.27	0.34	0.29	0.40	0.34	0.41	0.41	0.45	0.34	0.45	0.43	0.51
(4)	0,23	0,24	0,12	0,16	0,23	0,28	0,14	0,22	0,27	0,28	0,22	0,23	0,27	0,29	0,22	0,22
(5)	0,23	0,26	0,12	0,12	0,24	0,33	0,15	0,17	0,30	0,34	0,22	0,19	0,30	0,36	0,25	0,20
(6)	0.81	0.90	0.78	0.87	0.83	0.92	0.81	0.91	0.82	0.93	0.85	0.93	0.83	0.92	0.85	0.94
(7)	0.85	0.90	0.80	0.88	0.85	0.92	0.84	0.92	0.83	0.93	0.86	0.93	0.85	0.93	0.88	0.93
(8)	0.83	0.90	0.80	0.88	0.83	0.92	0.82	0.91	0.85	0.93	0.85	0.93	0.86	0.92	0.86	0.94
(9)	0.81	0.91	0.78	0.88	0.83	0.93	0.83	0.90	0.83	**0.94**	0.86	0.92	0.85	0.93	0.87	0.93
(10)	0.83	0.89	0.79	0.82	0.83	0.92	0.81	0.85	0.85	0.93	0.82	0.86	0.86	0.93	0.84	0.85
(11)	0.81	0.88	0.80	0.88	0.82	0.91	0.82	0.90	0.80	0.92	0.88	0.92	0.77	0.92	0.86	0.94
(12)	0.77	0.78	0.68	0.68	0.76	0.81	0.73	0.71	0.74	0.85	0.75	0.74	0.73	0.86	0.78	0.73
With our approach:																
(4)*	0.68	0.77	0.68	0.71	0.70	0.83	0.75	0.79	0.69	0.89	0.79	0.82	0.69	0.91	0.81	0.86
(5)*	0.60	0.71	0.60	0.64	0.65	0.80	0.62	0.66	0.68	0.83	0.67	0.66	0.67	0.84	0.64	0.69
(6)*	0.89	0.91	0.82	0.87	0.89	0.92	0.90	0.91	**0.95**	0.93	0.92	0.92	**0.95**	0.93	0.93	0.92
(7)*	0.90	**0.92**	0.84	0.88	**0.90**	0.92	0.91	0.91	0.94	0.93	0.92	0.92	0.94	0.92	0.92	0.92
(8)*	0.90	**0.92**	0.84	0.88	**0.90**	0.92	0.90	0.92	0.94	0.93	0.92	0.93	0.94	0.92	0.93	0.93
(9)*	**0.92**	**0.92**	0.85	0.88	**0.90**	0.93	0.92	0.92	0.94	0.93	0.92	0.92	0.93	0.93	0.92	0.92
(10)*	0.90	0.90	0.84	0.86	**0.90**	0.92	0.89	0.90	0.92	0.93	0.91	0.91	0.90	0.93	0.92	0.92
(11)*	0.88	**0.92**	0.85	**0.90**	0.87	**0.94**	0.90	0.91	0.92	**0.94**	0.92	0.93	0.93	0.93	0.93	0.93
(12)*	0.90	0.91	**0.88**	**0.90**	0.86	0.93	**0.93**	**0.93**	0.90	**0.94**	**0.93**	**0.94**	0.89	**0.94**	**0.95**	**0.95**

relationships between a class and its semantic neighbors (described in Section III-B). On the other hand, in the adaptation of path-based measures, we replace the shortest path function by our semantic distance function described in Section III-C. We use the * symbol to differentiate the adaptations from the originals. Also, we use these parameters values to keep the same importance of the evaluated parameters by the semantic measures.

In Table III, we present the F-score results of the evaluated ontology-based measures for the first category of the experiments. We evaluate each of these measures using window sizes of 2, 4, 6, and 8 for each dataset. In the top section of Table III, we present the results of the original approaches described in the state-of-the-art. In the bottom section, we show the results of our adaption in the feature and path-based measures. Based on the presented results, it is possible to note, in most of the cases, the adaptation of the measures archive better F-scores, in comparison with the original measures. Also, as described in Section II, the IC-based measures have an extreme disadvantage with a domain ontology where non-taxonomic relationships have a greater impact on WSD. In the feature-based approaches, we obtain a better set of features using the semantic measures adapted with our approach than the original approaches. Thus, we obtain better discrimination between the ontology classes. Finally, in this experiment, the path-based measures are very efficient to evaluate the relatedness value between two ontology classes, as well as the path-based measures are fundamental for more significant discrimination between the ontology classes in comparison with the other semantic measurement approaches.

Since path-based measures have better F-score result in WSD, for the second and third experiments, we use the semantic measure (6) (according to Table II). The measure (6) presents the simplest mathematical formulation, and this measure can demonstrate all the advantages and disadvantages regards the memory consumption and query time of the path-based measures. To evaluate the memory consumption and query time, we use the semantic measure (6) in three different approaches. In the first approach (I), we use the semantic measure (6) to calculate the relatedness values between all ontology classes before a query, and we store all these values in memory. In the second approach (II), the relatedness value between two ontology classes is calculated during the query time using the semantic measure (6) with the Dijkstra algorithm [20] as the distance function. In the third approach (III), the relatedness value between two ontology classes is calculated during the query time using the adaptation of semantic measure (6) with our proposal as the distance function (semantic measure (6)* in Table III).

In the second experiment, we evaluate the average number of classes in memory for each ontology class. The approaches (I) and (II) represent the two extremes of this experiment. The approach (I) requires, in memory, the relatedness values between all ontology classes, then this approach has as the average number of stored classes for each ontology class is equal to the number of the classes in ontology. In the approach (II), it is not necessary to store any additional reference for some ontology class, i.e., the average number of stored classes for each ontology class is equal to 0. With our approach (III), we obtain a considerable reduction in the average number of stored classes for each ontology class compared with the approach (I), achieving an average of 115

classes for each ontology class, considering all the classes of the Strataledge® ontology.

In the third experiment, we compare the average query time of the approaches (I), (II), and (III) in relatedness evaluation between all classes of Strataledge® ontology. This experiment is performed using a machine with an Intel i7-8700 CPU (3.2GHz) and 32GB of RAM on Windows 10. According to this experiment, the approach (II) has an average query time of 41 seconds, while our approach (III) has an average query time of 0.2 milliseconds. Intuitively, the average query time of the approach (I) is instantaneous. With this experiment, its possible to note that the Dijkstra algorithm [20] has poor efficiency in computing the distance between the Strataledge® ontology classes, in query time.

Although our approach (III) requires more memory than approach (II) and more query time that approach (I), our approach, in general (see semantic measure (6)* in Table III), obtain better F-score in WSD than the approaches (I) and (II) (see semantic measure (6) in Table III). Overall, we can say that our proposal is an alternative to the relatedness evaluation for ontologies that present non-taxonomic relationships with high impact on the discrimination of two ontology classes. Also, we are keeping the memory consumption and query time relatively low, with the advantage of improving, in most cases, the F-score results in WSD.

V. Conclusion

In this paper, we propose an approach to improve the path-based semantic relatedness calculation in query time. Firstly, we get all the relationships from an ontology class and its semantic neighbors through the semantic direct paths and store in memory. Then, in relatedness evaluation query, we use a semantic distance function to directly access the distance value between two ontology classes related through the direct semantic path or by calculating this value, in query time, through the indirect semantic paths.

To evaluate our approach, we perform the word sense disambiguation (WSD) of the terms that name different ontology classes. For this, we extract the sentences that these terms occur from four different datasets about geology in Oil&Gas domain. The experimental results show that our approach presents a better F-Score in WSD in most of the cases. Also, our approach drastically reduces the total number of classes in memory as well as keep a lower query time considering the traditional path-based measures. For future work, we will exploit weight techniques in order to have better distinguish between the ontology classes related through non-taxonomical relationships as well as split the HR-type relationships semantically.

Acknowledgment

This research was supported by Brazil Federal Agencies CNPq and CAPES, and the Petrobras company. We gratefully thank Geoscience Institute of UFRGS for the datasets and the Endeeper Company for the Strataledge® ontology.

References

[1] B. T. McInnes and T. Pedersen, "Evaluating measures of semantic similarity and relatedness to disambiguate terms in biomedical text," *Journal of biomedical informatics*, vol. 46, no. 6, pp. 1116–1124, 2013.

[2] S. Patwardhan, S. Banerjee, and T. Pedersen, "Using measures of semantic relatedness for word sense disambiguation," in *International conference on intelligent text processing and computational linguistics*. Springer, 2003, pp. 241–257.

[3] G. Hirst, D. St-Onge *et al.*, "Lexical chains as representations of context for the detection and correction of malapropisms," *WordNet: An electronic lexical database*, vol. 305, pp. 305–332, 1998.

[4] A. Lorenzatti, M. Abel, B. R. Nunes, and C. M. S. Scherer, "Ontology for imagistic domains: Combining textual and pictorial primitives," in *Advances in Conceptual Modeling - Challenging Perspectives*, C. A. Heuser and G. Pernul, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 169–178.

[5] S. Harispe, S. Ranwez, S. Janaqi, and J. Montmain, "Semantic similarity from natural language and ontology analysis," *Synthesis Lectures on Human Language Technologies*, vol. 8, no. 1, pp. 1–254, 2015.

[6] R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and application of a metric on semantic nets," *IEEE transactions on systems, man, and cybernetics*, vol. 19, no. 1, pp. 17–30, 1989.

[7] Y. Li, Z. A. Bandar, and D. Mclean, "An approach for measuring semantic similarity between words using multiple information sources," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 871–882, July 2003.

[8] X. Liu, Y. Zhou, and R. Zheng, "Measuring semantic similarity in wordnet," in *2007 International Conference on Machine Learning and Cybernetics*, vol. 6, Aug 2007, pp. 3431–3435.

[9] Z. Zhou, Y. Wang, and J. Gu, "A new model of information content for semantic similarity in wordnet," in *Future Generation Communication and Networking Symposia, 2008. FGCNS'08. Second International Conference on*, vol. 3. IEEE, 2008, pp. 85–89.

[10] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, ser. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 448–453.

[11] Y. Cai, Q. Zhang, W. Lu, and X. Che, "A hybrid approach for measuring semantic similarity based on ic-weighted path distance in wordnet," *Journal of Intelligent Information Systems*, vol. 51, no. 1, pp. 23–47, 2018.

[12] N. Seco, T. Veale, and J. Hayes, "An intrinsic information content metric for semantic similarity in wordnet," in *ECAI*, vol. 16, 2004, p. 1089.

[13] G. Pirró and J. Euzenat, "A feature and information theoretic framework for semantic similarity and relatedness," in *The Semantic Web – ISWC 2010*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 615–630.

[14] A. Tversky, "Features of similarity." *Psychological review*, vol. 84, no. 4, p. 327, 1977.

[15] S. Likavec, I. Lombardi, and F. Cena, "Sigmoid similarity - a new feature-based similarity measure," *Information Sciences*, vol. 481, pp. 203–218, 2019.

[16] E. G. M. Petrakis, G. Varelas, A. Hliaoutakis, and P. Raftopoulou, "X-similarity: Computing semantic similarity between concepts from different ontologies," *Journal of Digital Information Management (JDIM*, vol. 4, 2006.

[17] D. Diefenbach, R. Usbeck, K. D. Singh, and P. Maret, "A scalable approach for computing semantic relatedness using semantic web data," in *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics*, ser. WIMS '16. New York, NY, USA: ACM, 2016, pp. 20:1–20:9. [Online]. Available: http://doi.acm.org/10.1145/2912845.2912864

[18] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1994, pp. 133–138.

[19] D. Hao, W. Zuo, T. Peng, and F. He, "An approach for calculating semantic similarity between words using wordnet," in *2011 Second International Conference on Digital Manufacturing & Automation*. IEEE, 2011, pp. 177–180.

[20] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.