Journal of the Brazilian Computer Society, 2009; 15(1):27-43. ISSN 0104-6500

Ontology matching for dynamic publication in semantic portals

Fernanda Aparecida Lachtim, Ana Maria de Carvalho Moura*, Maria Cláudia Cavalcanti

Department of Computer Engineering, Military Institute of Engineering – IME,
22290-270, Rio de Janeiro, RJ, Brazil

Received: September 18, 2008; Accepted: March 28, 2009

Abstract: Semantic portals are characterized for storing and structuring content according to specific domain ontologies. This content is represented through ontological languages, which enable not only adding semantic value to information treatment, but also inferring new knowledge from it. Publication in a semantic portal is typically done by instantiating its ontology, and this is often performed manually or through the use of specific forms. However, in order to keep portals constantly up-to-date, it is necessary to provide means for a more dynamic publication, integrating the portal content with information retrieved from different ontology-based sites on the same or on complementary domains. Reusing information from different ontologies requires specific and efficient mechanisms to align them, taking into account syntactical and semantical conflicts. This paper proposes an extension of the Crosi Mapping System, a matching mechanism which calculates similarities between ontologies. Some of its original algorithms have been enriched with additional functionality. This extension, named *e-CMS*, has been evaluated using the OAEI ontology alignment benchmark, and results show an increase of 69% in alignment precision when compared to the CMS original version. In order to illustrate its use, the *e-CMS* strategy was applied to SiGePoS, a System for Generating Semantic Portals. The semantic module, one of the system components, implements the alignment mechanism between ontologies, which is performed by the *e-CMS*.

Keywords: *ontology matching, semantic portals, ontology interoperability, information integration.*

1. Introduction

Due to the heterogeneity and exponential growth of Web information, current traditional portals face difficulties in dealing with page maintenance. They are still very limited concerning mechanisms for exchanging, reusing and integrating content of other portals, and they rarely present efficient information retrieval strategies and metadata maintenance. Semantic portals emerge as an evolution of traditional portals[1, 24, 27, 28]. They use ontology as a means to provide added semantic expressivity to their information content, as well as to improve some of their functionalities, such as search, organization, sharing, publication and inference. The hierarchical structure provided by ontologies facilitates the organization and management of portal content, as well as allows for dynamic content publication. It also provides the user with a more integrated view of the information that he/she can be interested in, including the benefits of automatic deduction of new information.

A major benefit provided by semantic portals concerns the use of thesaurus or domain specific vocabularies integrated to ontologies to provide context. Context is any information that can be used to characterize the situation of an entity[4]. Thus, in those portals, such artifacts enable semantic contextualization, an important issue for the search process and dynamic instantiation in portals. For example, a search expression containing a specific ontology concept can be extended with additional terms related to that concept, be it by inheritance or by association, providing a more semantic and contextualized search.

Associated with the contextualization approach, emerges the multi-facet search paradigm proposed by Ranganathan[34] to facilitate navigation and provide more precise results for searches in the portal. The multi-facet paradigm consists of a more flexible approach for organizing items in a Web site, i.e., instead of categorizing items in a single classification, they can be annotated into multiple facets simultaneously, according to their meaning[29]. Recently, this idea has been largely explored in semantic portals[38], adding major benefits to their ontological structures and inference capacity.

However, despite major advances concerning functionalities in semantic portals, some important limitations still remain, due to the complexity of dealing with the heterogeneity and dynamicity of the Web. For example, since portals aim to be information reference sites, they should be kept very up-to-date to all the related sources of information, including Web sources. In order to keep them constantly up-to-date, it is necessary to provide mechanisms to support dynamic content publication in portals.

The semantic gap between traditional and semantic portals became more evident in a recent study[21,28], where content publication, navigation, organization and content

*e-mail: anamoura@ime.eb.br

management represent an important challenge, specially when these tasks are considered to be performed dynamically. Table 1 presents some of the main tasks usually provided by a portal, and shows how each one is currently performed in a traditional portal, and how it might be executed in a semantic portal in a near future.

In this context, consider for example, the semantic portals generators such as OntoWeb[15] and ODESeW[3]. Despite adopting ontologies as the main approach to organize and interoperate information, they do not provide an automatic mechanism capable of interoperating ontologies and publishing information dynamically.

However, generating information dynamically for a semantic portal is not a trivial task. It requires interoperability techniques that enable aligning (semantic) Web documents or ontologies with the portal domain ontology, so that only relevant information is classified and instantiated. Due to the importance of these techniques, we introduce this paper with a brief overview of ontologies interoperability with emphasis on some matching systems. This study was essential in this work, since it motivated an extension of the Crosi Mapping System[18], one of the matching mechanisms studied calculates similarities between ontologies. Some of the CMS original algorithms have been enriched with additional functionalities in order to provide better results within our system.

This extension, namely *e-CMS*, has been incorporated into a system that provides the basic infrastructure to dynamically generate content for a semantic portal, in order to enable integration, organization, and publication of information based on the intensive use of ontologies.

Therefore, this paper has two main contributions: i) to describe the *e-CMS* functionalities and show how this extension improved its metric results using an ontology benchmark as a test bed; and ii) to illustrate how this strategy has been implemented in the SiGePoS, a System for Generating Semantic Portals. The main component of SiGePoS, the semantic module, is responsible for retrieving distributed information on the Web according to the domain ontology that sustains the portal. In this work we describe how mappings are established between this information and that domain ontology.

SiGePoS represents the application of the *e-CMS* in an educational semantic portal, and shows, in practice, how ontology instances can be dynamically categorized and published in the portal.

Table 1. Main differences between traditional and semantic portals - adapted from Reynolds et al.[35].

Task	Traditional Portals	Semantic Portals
Search	Based on free text	Based on ontologies
Navigation/ organization	Fixed hierarchy of classification	Multidimensional with facets
Content management	Central management	Towards a decentralized management
Processing	Non-structured text	Done by agents, according to oriented-structured text
Content publication	Usually manual or by means of specific forms	Automatic, through ontology integration

The rest of the paper is described as follows. In Section 2 we give a brief overview of the main approaches found in the literature for interoperating ontologies, including some related work on matching systems. Section 3 describes the CMS (Crosi Mapping System) and its extended version, the *e-CMS*, with full details of the additional functionalities it has incorporated. Section 4 presents an evaluation of the proposed *e-CMS*, using an appropriate benchmark that contains a rich set of ontologies. Section 5 describes the the SiGePoS system with a brief description of its main components. This system has been developed according to an architecture proposed to generate content for semantic portals. That section also presents a case study in order to demonstrate the SiGePoS usage. Finally, Section 6 concludes the paper, with suggestions for future work.

2. Ontologies Interoperability and Related Work

Ontologies represent the central key to the Semantic Web[13]. They are responsible for providing the necessary semantics to contextualize information, enabling interoperability across heterogeneous systems and semantic web applications. Nevertheless, interoperating information on the Web is a complex task, since much of information is described using natural language, without any metadata associated. Creating rich and ontology-based metadata is one of the major challenges in developing the Semantic Web, and it seems to be the solution to allow mechanisms to adequately interoperate ontologies.

(Re)using information from different ontologies requires specific and efficient mechanisms, which should be able to cope with distinct levels of interoperability[9]. Four different approaches to deal with ontologies interoperability are proposed in the literature: mapping, alignment, merge and integration[2].

Ontology mapping determines when two ontologies are semantically related at conceptual level, and how instances of the source ontology are transformed into instances of the target ontology according to its semantic associations, also providing mechanisms to transfer instances among them[36, 26]. More oriented to deal with complementary ontologies, the alignment approach results in a mutual accordance, where two source ontologies become consistent and coherent, generating as a final result the two original ontologies separately, but with additional links between their equivalent terms. Merge is the process of unifying ontologies of similar domains, where versions of the original ontologies are merged into a single one, with all their terms together, and without referencing their origins.

Finally, in the integration approach, three perspectives can be considered[33]:

i) the merge approach previously described;

ii) when an application uses concepts from one or more ontologies; and

iii) when a domain ontology is created reusing one or more ontologies on different domains. The reuse process here is characterized as a result of generalizing, specializing or adapting concepts from other existing ontologies. The latter perspective is the one considered in the scope of this work.

These approaches can be considered as an ontology reuse process, and they show different levels of commitment when matching ontologies: *alignment* < *mapping* < *integration* < *merge.* The *alignment* process establishes only correspondences between ontologies entities, while in the *mapping* process there is an infrastructure to transfer instances from the source ontology into the target one. The *integration* process creates a new ontology from the reuse of others, although it keeps the references to the source ontologies. Finally, the *merge* process generates a new ontology, even though it does not preserve any link with the original ontologies.

However, all these approaches are preceded by a technique usually called *matching*[6, 14, 16, 18], although sometimes also referred to as an alignment[7, 8, 16, 17, 20] or even as a mapping process[30, 31]. Regardless of name, this technique consists in comparing terms from a selected ontology with the terms from a source ontology, so that measurements of lexical, structural and semantic similarities can be established between these terms. Thus, the more refined the similarity results calculated by this process, the better the results provided by the interoperability mechanisms. In this work we adopt the term *matching* to denote this technique.

In recent years many systems have emerged to automate the matching process[6], although using different strategies and algorithms. ASMOV[16], Falcon-AO[14], Ontodna[20], DSSim[30, 31], CMS (CROSI Mapping System)[18] are just some examples of tools included in this category of systems. Another list of matching tools can also be found in[6, 8].

Considering that the dynamic generation of content into a semantic portal depends on ensuring interoperability between (semantic) Web ontologies and the portal ontology, a deep study of these matching tools became essential.

2.1. Matching tools

In this section we give a brief overview of some matching tools found in the literature[16, 14, 20, 30, 31, 18]. They all claim to provide interoperability between ontologies and participate in the OAEI[I] (*The Ontology Alignment Evaluation Initiative*), an initiative created to evaluate this kind of tools. This overview does not intend to be exhaustive, but it focuses on those approaches that present the important features required for our work, considering the techniques employed to deal with syntax, structural and semantic heterogeneities normally found during the matching process between ontologies.

ASMOV[16] is an automatic ontology matching tool designed for ontology integration. It produces mappings between concepts, properties and individuals, including mappings from object properties to datatypes properties and vice-versa. Similarity measures are iteratively calculated between entities for a pair of ontologies with the help of thesaurus (Wordnet and UMLS) and the following features are analyzed: lexical description; external structure (parents and children), internal structure (properties restrictions for concepts, types, properties domains and ranges and data values) and individual similarity. Similarity matrices and graphs are generated to provide structural and semantic validation, since invalid mappings are detected.

Falcon AO[14] is another automatic ontology matching system to interoperate semantic Web applications that use related ontologies expressed in RDF and OWL. It includes the following features: linguistic matching, by means of virtual documents (a collection of weighted words related to a class or a property) containing local description and information about the meaning of the entity; structural matching, which employs RDF bipartite graphs to compute structural similarities between domain entities and statements; and semantic analysis that is done over block of mappings (clusters) composed of the domain entities of each ontology, according to their structural proximity. Rules are employed to eliminate semantic conflicts, complementing this phase by tuning up thresholds based on measures obtained in the linguistic and structural phases.

OntoDNA[20] goes further than the first two tools, providing automatic ontology mapping and merging. It uses data mining methods and clustering techniques incorporated with lexical similarity to perform the different phases of the process. It uses Formal Concept Analysis – FCA[12], to capture the ontology structure and properties; and Self-Organizing Map – SOM[39] and K-Means[25] to process structural and semantic heterogeneities between ontologies. Lexical heterogeneity is resolved by means of an edit distance technique with a threshold value of 0.8 to discover lexical similarity.

Differently from the other mentioned systems, DSSim[30, 31] was conceived based on the fact that ontology mappings contain inconsistencies, missing or overlapping elements and different entity meanings, hence introducing a certain amount of uncertainty into the process. In order to cope with these problems, this system adopted a multi-agent architecture, where each agent builds up a belief for the correctness of a particular mapping hypothesis. Beliefs or similarity assessments are the comparison results between all concepts and properties of two ontologies (the WordNet dictionary is used in this mapping process). These beliefs are stored into matrices and, after eliminating inconsistencies, they are combined into a more coherent view, in order to provide more refined mappings. It uses a specific technique (Dempster-Shafer theory of evidence[37]) to handle missing data, as well as to model and reason uncertain information. This technique has recently incorporated a multiword ontology entity labels to provide compound term comparisons and abbreviations based on defined language rules[31].

Last but not least, CMS[18] is yet another automatic matching ontology system. Similarly to the previous tools, it captures OWL ontologies and matches them with the aid of external linguistic resources. CMS implements mapping techniques as independent components, namely: name matchers

[I] http://oaei.ontologymatching.org/2007/benchmarks/

and semantic matchers. Name matchers are oriented to solve lexical and syntactical heterogeneities using thesaurus and edit-distance functions, while semantic matchers add a semantic flavor to the process. Heuristic rules are used to exploit structural information, identifying class hierarchies and properties between ontologies. CMS API was publicly available sooner than the other matching tools. Its modular architecture facilitates code reuse and extension. A more detailed description of the CMS system is given in section 3.

2.2. Discussion

These tools share many common features: they all process OWL and RDF large-scale ontologies, and provide distinct and iterative phases to perform lexical, syntactical and semantic heterogeneities, supported by dictionaries and thesaurus. Their major difference lies in the algorithms used to perform all these matching phases. Some use lexical comparison and statistic analysis to obtain linguistic similarities, and graphs and similarity matrices[16] in combination with rules to analyze ontology structure[14], whereas CMS[18] employs more sophisticated functions to implement string distance. OntoDNA[20] adopts a different approach based on data mining and clustering techniques, while DSSim[30] is the only tool that considers partial knowledge in its ontology mapping method. It is able to represent uncertainty, and establishes a set of hypotheses to handle with uncertain information.

Euzenat et al.[8] present a detailed evaluation of these matching tools after being submitted to a large variety of test cases, which consisted of different tracks such as benchmark, expressive ontologies (anatomy) and directories (environment, food, etc.). All these tests have been evaluated according to standard evaluation measures, i.e., precision and recall, which have been computed against the reference alignments. The results of the OAEI initiative emphasize the importance and the maturity level these achieved over the last three years. The fact is that no system has had the best performance in all the tests. For example, ASMOV obtained the best performance in the benchmark track, while Falcon-AO performed best for the thesaurus merging scenario, so far considered as one of the best ontology matching systems.

However, a very important issue in this study concerned the availability of a matching system as an open API that could be incorporated to our architecture, since developing matching systems was beyond our scope. The current work started at the end of 2006, and then, after an exhaustive investigation in the literature on this subject, CMS was the only open code available as an API. It is worth mentioning that, nearly an year later, motivated by the OAEI workshops (2007 and 2008), some systems became available for download (ASMOV[II], Falcon-AO[III], OntoDNA[IV], DSSim[V]). At that point we had already decided to extend CMS with the goal of implementing the *SiGePoS* system.

Furthermore, considering the OAEI results obtained in 2007[8], and the CMS OAEI results obtained in 2005[19] (both combined in Table 2), we observe that CMS shows a performance, regarding precision, very close to the other systems. It also shows similar results with respect to the F-measure, for groups 1xx and 2xx.

3. CMS and its Extended Version

The *CMS* mechanism has been chosen to support the matching process of the semantic module of the SiGePoS system, which will be presented later in section 5. Nevertheless, during the development of the system, it was possible to improve some results in the similarity calculation process, between the domain ontology (the target ontology) and the source ontology retrieved from the Web, since important aspects to our application were not being considered by CMS. Hence, this section aims at describing the main adaptations introduced in the CMS algorithm, considering at first some details of its original version, prior to introducing its extended version.

3.1. Some issues of the CMS algorithm

The CMS calculates entity similarity between two ontologies: (source and target). In this paper we consider property and relationship as equivalent terms.

Similarity calculation is based on the edit distance algorithms of the *second string* packet[VI], and includes different classes to process this task, such as[18]:

- *OntWNMatcher*: matches entity names using WordNet;

[II] http://support.infotechsoft.com/integration/ASMOV/OAEI-2008/, available upon request.

[III] http://iws.seu.edu.cn/projects/matching/

[IV] http://pesona.mmu.edu.my/~cckiu/OAEI2007.htm

[V] http://pesona.mmu.edu.my/~cckiu/OAEI2007.htm

[VI] This package, developed by researchers of Carnegie Mellon University, is an open source code built in Java to deal with string matching. It consists of several algorithms, such as: *Levenshtein distance, Monger-Elkan, Jaro, Jaccard similarity, Jensen-Shannon*, among others.

Table 2. Results obtained by participants on the benchmark test case (corresponding to harmonic means). OAEI tests involve comparing and matching a set of ontologies with a reference ontology. These tests are organized in three main groups: simple tests (1xx), such as comparing the reference ontology with itself; systematic tests (2xx), which consists of discarding features from the reference ontology and matching the modified and original ontologies; and real tests (3xx), which involve matching real ontologies to the reference ontology.

Algo	ASMOV			DSSim			Falcon			OntoDNA			CMS-MC		
	Prec	Rec	F-mea	Prec	Rec	F-mea	Prec	Rec	F-mea	Prec	Rec	F-mea	Prec	Rec	F-mea
1xx	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,94	1,00	0,97	1,00	1,00	1,00
2xx	0,95	0,90	0,92	0,99	0,60	0,75	0,92	0,85	0,88	0,80	0,43	0,56	0,91	0,45	0,60
3xx	0,85	0,82	0,83	0,89	0,67	0,76	0,89	0,79	0,84	0,90	0,71	0,79	0,96	0,42	0,58

- *OntWNPlusMatcher*: the same as before, except that it also uses string distance algorithms;
- *OntWNDisSimMatcher*: matches names using WordNet structure similarity;
- *OntStructurePlusMatcher*: matches concepts based on their definitions and location in the class hierarchies;
- *OntCanoNameMatcher*: matches names in canonical forms;
- *OntHierarchyDisSimMatcher*: matches concepts based on the structure similarity with regard to class hierarchy.

These classes can be combined as a sequence previously defined, generating aggregated similarity values. The similarity calculation process is carried out when all the terms (entity names) of a source ontology are compared with those of the target ontology, as a cartesian product.

Calculation of the neighbor terms is done by analysing the relationships, their domains and ranges using edit distance algorithms, without considering any additional knowledge support, such as vocabularies or synonym dictionaries. It also uses these algorithms to verify the ancestors of an entity, although it does not consider its descendants.

Due to our application requirements, two classes have been identified as possible candidates for extension from the list above: *OntStructurePlusMatcher* and *OntCanoNameMatcher*. The first for dealing with properties, and the second for considering hierarchies. Their main algorithm steps are described next:

OntStructurePlusMatcher

i. Matches terms using edit distance algorithms (*second string*);
ii. Stores previous similarity calculation as the initial one;
iii. Retrieves all related relationships;
iv. Gets all the domain and ranges of each relationship;
v. Matches domains and ranges with edit distance algorithms;
vi. Uses the similarity of the relationships domains and ranges to refine the initial similarity.

Furthermore, heuristics are included in this process to calculate similarity of ontology properties. An arbitrary weight (w) is assigned to a property, according to the following situations:

- (PDR): if property, domain and range are equivalent , then w = 1;
- (PD): if property and domain are equivalent, then w = 0.9;
- (PR): if property and range are equivalent, then w = 0.8;
- (DR): if domain and range are equivalent, then w = 0.7;
- (D): if domain is equivalent, then w = 0.4;
- ( ): if nothing is equivalent, then w = 0.

OntCanoNameMatcher

This class takes the canonic name of an entity, is represented by the class name concatenated to the local name of its parent class, or if it does not have a parent, it is concatenated to its descendant nodes. For example, if A is subclass of B, which is subclass of C, then the canonical names of A and B are C.B.A and C.B respectively. Thus, terms are extracted from the source and target ontologies, generating canonical names, before being submitted to the edit distance algorithms. The sequence order of execution within this class is:

i. Select the terms with a low similarity value (different terms);
ii. Retrieve the ascendants of these terms;
iii. Concatenate these terms to their ascendants;
iv. Match each term of a concatenated sequence of a source ontology with the one within the target ontology.

3.2. *The extended CMS (e-CMS)*

In order to obtain a more refined matching process, able to cope with our system requirements, we decided to extend the original version of CMS, taking into account the flexibility of the API code available to users. The idea was to aggregate into CMS some specific strategies considered important in the matching process for our system, in order to bring benefits and richer results from the similarity calculation process. The new extended version of CMS was named *e-CMS*.

This extension contemplated some important issues, as described next.

3.2.1. Algorithm adaptations

The first modification in the CMS algorithm concerned the implementation of a mechanism, namely *oriented matching*. This strategy makes it possible that only a subset of terms within the source ontology needs to be compared with some terms within the target ontology. As it will be described later in section 5, source ontologies correspond to documents that are retrieved from queries submitted to a semantic query search machine. These documents probably contain terms and relationships that are included in the queries, representing therefore, a subset of the ontology entities if we consider the whole ontology. Hence, we adapted the *directed matching* strategy not only by modifying the weight values in the similarity calculation, but also by considering only the entities included in the queries.

The second modification concerns the extension of the following API Java classes:

- *OntStructurePlusMatcher*: Besides calculating the entities similarity by using the *edit distance* algorithm, a domain dictionary has been added to this calculation, in order to improve precision and recall. In the scope of this work a dictionary on the educational domain was created with equivalent terms for concepts and relationships, both in English and in Portuguese. In its original version, this algorithm verified only the relationships that included some kind of restriction. Hence, this modification made it possible to analyze a relationship, independently of any restriction;

- *OntoCanoNameMatcher:* The domain dictionary mentioned previously was also added to this process as an extension to this class. Furthermore, this class considered only the concepts hierarchies which were different in the source and target ontologies, i.e., when matching two concepts of these ontologies, if they had the same name, they were considered similar, independently of their context. Thus, we added to this procedure an analysis of all the concepts hierarchies in order to consider their context neighborhood. Hence, the analysis of the concept descendants, not considered before, has been added to this *API*, since only the ascendant concepts were taken into account in its original version.

Finally, we changed the sequence of execution in the CMS algorithm, as showed in Figure 1. It comprehends four main steps:

i. Analysis of terms similarity between the source and target ontologies, using the edit distance algorithms;

ii. If terms are not similar, their equivalent terms are also analyzed;

iii. Even if the terms above are considered similar, their relationships (domain and range) are verified to ensure

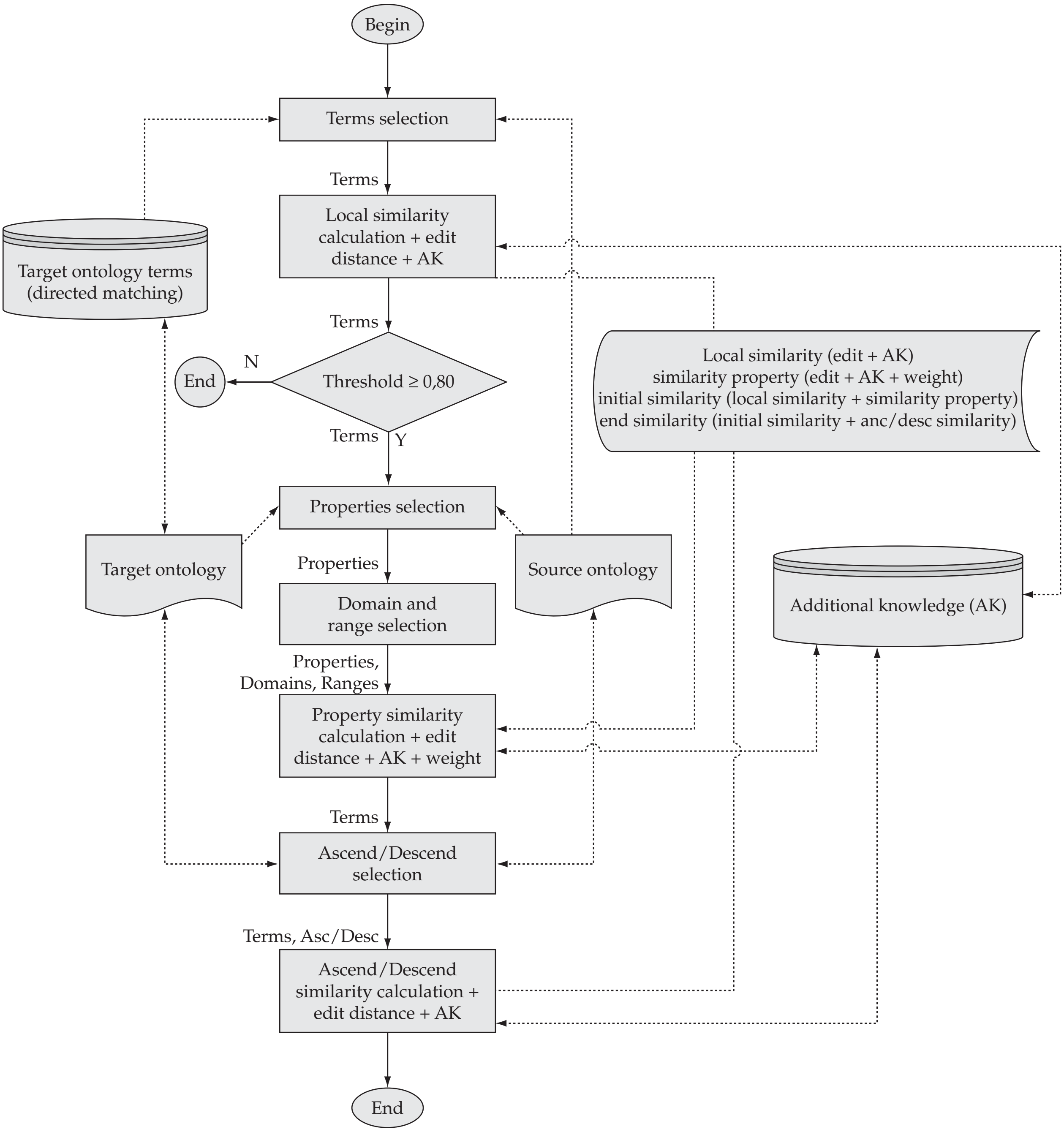

Begin
Terms selection
Terms
Local similarity calculation + edit distance + AK
Target ontology terms (directed matching)
Terms
End
N
Threshold ≥ 0,80
Terms
Y
Local similarity (edit + AK)
similarity property (edit + AK + weight)
initial similarity (local similarity + similarity property)
end similarity (initial similarity + anc/desc similarity)
Properties selection
Target ontology
Properties
Source ontology
Additional knowledge (AK)
Domain and range selection
Properties, Domains, Ranges
Property similarity calculation + edit distance + AK + weight
Terms
Ascend/Descend selection
Terms, Asc/Desc
Ascend/Descend similarity calculation + edit distance + AK
End

Figure 1. The *e-CMS* algorithm execution order.

they correspond to the same context. Equivalent terms of these properties are also considered in this step;

iv. Finally, ascendant and descendant concepts present in the source and target ontologies are also analyzed using the edit distance algorithms, aided by an additional knowledge, i.e., a domain dictionary. The analysis of neighbor terms that takes into account properties, ascendant and descendant concepts are very important to ensure contextualization.

4. Experimental Tests Using the OAEI Benchmark

The *e-CMS* evaluation was conducted using a benchmark and an *API module* originally proposed to evaluate ontology alignment mechanisms. Both, benchmark and API, are included in the OAEI initiative, mentioned in section 2.1. This benchmark includes a collection of 51 ontologies on the bibliographic references domain, from which *precision, recall and F-measure*[VII] values have been measured during the similarity calculation process of a selected set of mechanisms, including the original *CMS* and the proposed *e-CMS*.

This evaluation consisted of a set of tests, where each test compares a source ontology (reference ontology, which is a complete ontology on bibliographic domain), with the remaining 50 ontologies (test ontologies), most of them created as a modified version of the reference ontology. These modifications contemplated basically:

i. modification of entity names, that may be replaced by random strings, synonyms, names in other languages, etc.;

ii. comments, which can be eliminated or translated into other languages;

iii. hierarchies, which can be eliminated, extended or flattened;

iv. instances can be eliminated;

v. properties can be eliminated or restrictions in the classes can be suppressed; and

vi. classes can be extended or flattened.

The mechanisms that took part in these tests, generated a file for each ontology pair, in a format defined by the alignment *API*, containing for each matched concept pair, an associated weight value. These files were compared to a standard set of pre-defined similar files (*refAlign*) included in the benchmark, which contained all expected matched concept pairs (source and target) and the weight values defined as a reference for the tests. Finally, based on these latter files, another pair of files containing precision, recall[19] and F-measure values, respectively, was generated.

4.1. Experimental tests using the OAEI benchmark

Tests have been run using three alignment mechanisms, available in the benchmark to compute similarities: *RefAlign, SMO* and *Levenshtein.* Additionally, it included three other versions of *CMS*:

i. *CMS-MC*, that matches all the best similarity values obtained from the alignment mechanisms, selecting those with higher recall, precision and F-measure values;

ii. *CMS Strut-Cano*, which uses only two classes of the original CMS; and

iii. the *e-CMS*, with the extensions proposed in section 3.2. During the tests with the 51 ontologies, for all precision, recall and F-measure values obtained, a minimum threshold of 0.80 was considered for the similarity matching values, implying that matches under this threshold were discarded.

Table 3 presents the precision values, where it is possible to notice that *CMS-MC,* in most cases, presents the best values. This is due to the fact it contains the best similarities values previously and manually selected from exhaustive tests. On the other hand, *e-CMS,* in most cases (as compared to the other algorithms), presents the best results. Table 3 also shows the recall values, where *Levenshtein* and *e-CMS* present, in most cases, the best values.

OAEI tests with ontologies have been grouped according to their common characteristics, and were organized into three groups (1xx, 2xx, and 3xx) as listed in Table 2. In this analysis, we organized them into five groups (A-E). Groups A and E correspond to groups 1xx and 3xx, respectively. However, group 2xx was subdivided into 3 other groups: B, C and D. Each group consists of a set of tests (lines in Table 3), and each line corresponds to a test ontology. These groups are described as follows:

- Group A: Test ontologies in this group involve not only ontologies completely different from the reference ontology, as well as of other ontologies whose domain is similar to the reference one. The *e-CMS* shows best performance when compared to *CMS_strutCano*, due to the dictionary of terms which has been included in the matching algorithm, loosing only to *CMS-MC*, for the same reason pointed above;
- Group B: In this group, test ontologies involve entities whose names have been replaced by strings, synonyms and names in other languages, a fact that is well treated by *e-CMS*. *CMS-MC* has the best performance for precision, although *e-CMS* presents the best values for recall and F-measure;
- Group C: In the test ontologies from this group, properties have been added or eliminated, and hierarchies have been modified. Again *e-CMS* has the best values for recall and F-measure and *CMS-MC* has the best performance for precision;
- Group D: In the test ontologies from this group, class names, comments and labels have been eliminated and replaced by random strings, but some test ontologies maintain the same properties. Since *e-CMS* does not consider instances for similarity calculation, it presented the worst results in recall, precision and F-measure;
- Group E: The test ontologies in this group are characterized by differences in structure, class names

[VII] The F-measure (F) is defined as a harmonic mean of precision (P) and recall (R) values, i.e., F = 2PR / (P+R). This measure was derived by van Rijsbergen (http://www.dcs.gla.ac.uk/Keith/Preface.html).

Table 3. Precision, recall and F-measure results for different algorithms.

	SMO			**Levenshtein**			**CMS – MC**			**CMS – StrutCano**			**e-CMS**		
	Prec	**Rec**	**Fmea**	**Prec**	**Rec**	**Fmea**	**Prec**	**Rec**	**Fmea**	**Prec**	**Rec**	**Fmea**	**Prec**	**Rec**	**Fmea**
A	0,618	1,000	0,764	0,618	1,000	0,764	NaN	NaN	-	0,867	0,671	0,757	0,889	1,000	0,941
	0,000	NaN	-	0,000	NaN	-	NaN	NaN	-	NaN	NaN	-	NaN	NaN	-
	0,618	1,000	0,764	0,618	1,000	0,764	1,000	0,788	0,881	0,843	0,660	0,740	0,853	1,000	0,921
	0,618	1,000	0,764	0,618	1,000	0,764	1,000	0,788	0,881	0,843	0,660	0,740	0,853	1,000	0,921
B	0,016	0,011	0,013	0,017	0,011	0,013	1,000	0,189	0,318	0,646	0,207	0,314	0,650	0,269	0,381
	0,618	1,000	0,764	0,618	1,000	0,764	1,000	0,697	0,821	0,843	0,609	0,707	0,865	0,990	0,923
	0,594	0,949	0,731	0,549	0,763	0,639	1,000	0,605	0,754	0,831	0,609	0,703	0,836	0,897	0,865
	0,302	0,330	0,315	0,221	0,176	0,196	1,000	0,230	0,374	0,718	0,289	0,412	0,867	1,000	0,929
	0,367	0,382	0,374	0,334	0,310	0,322	1,000	0,255	0,406	0,605	0,269	0,372	0,624	0,361	0,457
	0,367	0,382	0,374	0,334	0,310	0,322	1,000	0,264	0,418	0,605	0,269	0,372	0,624	0,361	0,457
	0,594	0,949	0,731	0,549	0,763	0,639	1,000	0,473	0,642	0,800	0,454	0,579	0,817	0,722	0,767
	0,302	0,330	0,315	0,221	0,176	0,196	1,000	0,103	0,187	0,609	0,145	0,234	0,879	0,980	0,927
C	0,618	1,000	0,764	0,618	1,000	0,764	1,000	0,708	0,829	0,843	0,660	0,740	0,853	1,000	0,921
	0,961	1,000	0,980	0,961	1,000	0,980	1,000	0,788	0,881	0,843	0,660	0,740	0,853	1,000	0,921
	0,618	1,000	0,764	0,618	1,000	0,764	0,788	0,788	0,788	0,843	0,660	0,740	0,853	1,000	0,921
	0,359	1,000	0,528	0,363	1,000	0,533	0,788	0,788	0,788	0,000	0,000	-	0,718	1,000	0,836
	0,542	1,000	0,703	0,576	1,000	0,731	1,000	0,760	0,864	0,747	0,653	0,697	0,780	1,000	0,876
	0,618	1,000	0,764	0,618	1,000	0,764	1,000	0,788	0,881	0,843	0,660	0,740	0,853	1,000	0,921
	0,961	1,000	0,980	0,961	1,000	0,980	1,000	0,788	0,881	0,843	0,660	0,740	0,853	1,000	0,921
	0,359	1,000	0,528	0,363	1,000	0,533	0,838	0,788	0,812	0,000	0,000	-	0,711	0,970	0,821
	0,892	1,000	0,943	0,917	1,000	0,957	0,788	0,788	0,788	0,000	0,000	-	0,718	1,000	0,836
	0,949	1,000	0,974	0,959	1,000	0,979	1,000	0,724	0,840	0,843	0,689	0,758	0,846	1,000	0,917
	0,951	1,000	0,975	0,951	1,000	0,975	0,961	0,757	0,847	0,843	0,660	0,740	0,837	0,990	0,907
	0,319	1,000	0,484	0,330	1,000	0,496	0,766	0,793	0,779	0,091	0,035	0,051	0,675	1,000	0,806
	0,352	1,000	0,521	0,359	1,000	0,528	0,757	0,757	0,757	0,000	0,000	-	0,663	0,940	0,778
	0,892	1,000	0,943	0,917	1,000	0,957	0,838	0,788	0,812	0,000	0,000	-	0,711	0,970	0,821
	0,806	1,000	0,893	0,879	1,000	0,936	0,766	0,793	0,779	0,000	0,000	-	0,675	1,000	0,806
	0,825	1,000	0,904	0,869	1,000	0,930	0,757	0,757	0,757	0,000	0,000	-	0,663	0,940	0,778
D	0,016	0,110	0,028	0,017	0,011	0,013	-	-	-	0,100	0,011	0,020	0,097	0,011	0,020
	0,016	0,110	0,028	0,017	0,011	0,013	-	-	-	0,100	0,011	0,020	0,097	0,011	0,020
	0,000	0,000	-	0,000	0,000	-	-	-	-	0,000	0,000	-	0,000	0,000	-
	0,016	0,110	0,028	0,017	0,011	0,013	-	-	-	0,100	0,011	0,020	0,097	0,011	0,020
	0,016	0,110	0,028	0,017	0,011	0,013	-	-	-	0,100	0,011	0,020	0,097	0,011	0,020
	0,016	0,110	0,028	0,017	0,011	0,013	-	-	-	0,100	0,011	0,020	0,097	0,011	0,020
	0,000	0,000	-	0,000	0,000	-	-	-	-	0,000	0,000	-	0,000	0,000	-
E	0,873	0,787	0,828	0,938	0,246	0,390	1,000	0,363	0,533	1,000	0,033	0,064	1,000	0,263	0,416
	0,682	0,625	0,652	0,936	0,605	0,735	1,000	0,348	0,516	NaN	0,000	-	0,964	0,584	0,727
	0,662	0,837	0,739	0,887	0,796	0,839	1,000	0,474	0,643	0,858	0,490	0,624	0,843	0,796	0,819
	0,858	0,948	0,901	0,908	0,908	0,908	0,850	0,566	0,680	0,912	0,540	0,678	0,877	0,869	0,873

- and properties. They have not been created based on the reference ontology, since they represent real ontologies, on bibliographical domain. All *CMS* algorithms versions presented good results for precision values, while recall values presented smaller values as compared to the other groups. For most tests in this group, *e-CMS* presents the best results with respect to the F-measure values.

4.2. Tests using different thresholds

In order to conduct a better analysis of results using *e-CMS*, we executed the same benchmark tests of section 4.1, using different thresholds: 0.60, 0.80 and 0.95. Figures 2, 3 and 4 present, respectively, precision, recall and F-measure results for these tests.

When comparing precision results using threshold 0.60 to the others varying from 0.80 to 0.95, one notices a considerable reduction of values in all occurrences, due to the increasing number of matching entities, although many of them have been considered irrelevant. Although the higher values of precision present low occurrences with threshold 0.80, as compared to threshold 0.95, the former still presents more better results for recall. With respect to the F-measure, the results for the 0,80 threshold were mostly better than for those obtained with other tested thresholds.

Increasing the threshold to 0.95 resulted in very inexpressive alterations in tests involving ontologies in other languages and in tests with ontologies composed of random

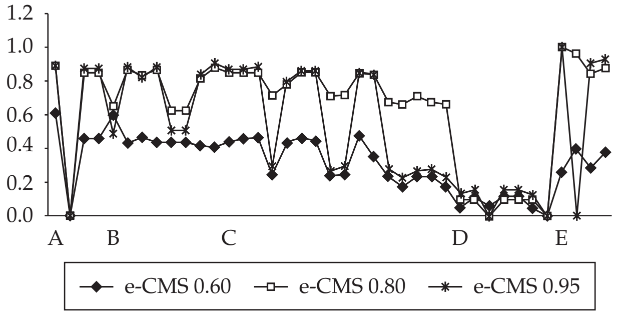


Figure 2. Precision results using different thresholds.

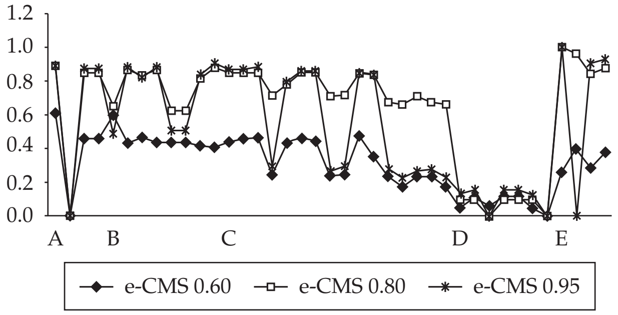


Figure 3. Recall results using different thresholds.

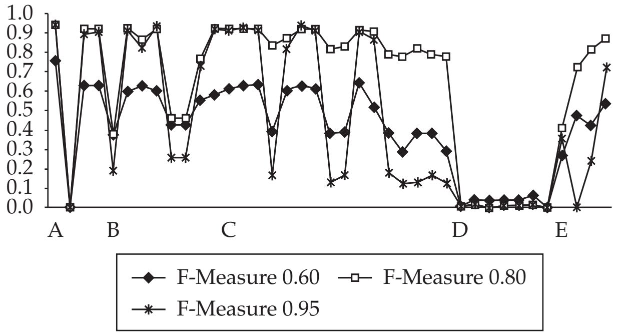


Figure 4. F-Measure results using different thresholds.

strings. Recall, precision and F-measure presented poor values in tests with 2 ontologies in group C, since contextualization properties in the hierarchy are very important in *e-CMS*, which does not take instances into account in the similarity calculation.

It is worth noticing that the higher the restriction level in the matching process, the lower the number of matching results. This is the case, for example, of tests with some ontologies of the groups C and D, which have had properties and comments suppressed and hierarchies modified. While there have been many matching results for thresholds lower than 0.95, this did not occur for threshold 0.95.

Figures 2 and 3 show that for threshold 0.80, precision values present 32% more better results as compared to threshold 0.95. However, recall results for threshold 0.80 present a set of better values, corresponding to 58% as compared to threshold 0.95. With respect to the F-measure, Figure 4 shows that for threshold 0.80 results present 85% of better values as compared to threshold 0.95, while the other 15% remain similar. For threshold 0.60 all results present lower values as compared to threshold 0.60.

Furthermore, no recall value for the latter threshold presents better results than for threshold 0.80. Hence, since the quality of the matching process depends from both precision and recall level, this analysis showed that threshold 0.80 is the best one, since it presents best results for the F-measure. Consequently, threshold 0.80 has been chosen as the most appropriate for similarity calculation in the *SiGePoS* system.

4.3. Tests considering properties, with different weights

As already mentioned in section 3.1, *CMS* uses specific weights (w) to calculate similarity of ontology properties. Seeking for better results in the recall, precision and F-measure results for *e-CMS* using threshold 0.80, we decided to modify the original weights when calculating property similarities. Three tests have been run, incrementing the original weights by 10, 20 and 30% respectively. However, these experiments did not produce good results, since all the recall, precision and consequently F-measure values were reduced. Hence we changed the strategy, applying the opposite condition, i.e., decreasing the original weights in 10, 20 and 30% respectively. In this case the results provided by the *e-CMS* improved with an increase on precision values.

Figures 5, 6 e 7 present the new values for precision, recall and F-Measure, setting the weight parameters (PDR, PD, etc., as described in section 3.1) with values 1, 0.80, 0.70, 0.60, 0.30 and 0, respectively. Comparing the graphs on Figures 2, 3 and 4 with those on Figures 5, 6 and 7, respectively, we notice that precision, recall and F-measure decreased for some ontologies of group B and for others either there was a very subtle recall decrease or the same value was maintained.

Further, this comparison shows that 69% of precision results have increased, 8% have decreased and 24% remained stable, while for recall results 61% have remained the same, with a subtle decrease for the other values. With respect to the F-measure values (comparing Figures 4 and 7), we can observe an increase of 38% and a decrease of 35%, while 25% of the values remained constant.

4.4. Discussion

From the previous analyses it was possible to conclude that, despite having a decrease for recall results, precision results presented an increase of 69%. Analyzing F-measure values we notice a subtle increase on the results. Therefore, we can conclude that it is worth using the *e-CMS* with this score modification (decrease of 10%), for the threshold 0.80.

Even though our work has been conceived strongly based on the CMS algorithm, and showed gains when compared to all its flavors, it is interesting to compare our results with related works, mentioned in section 2. Table 4 extends Table 2

Table 4. Adaptation of Table 2, with e-CMS results.

Algo	ASMOV			DSSim			Falcon			OntoDNA			CMS-MC			e-CMS		
	Prec	Rec	F-mea	Prec	Rec	F-mea	Prec	Rec	F-mea	Prec	Rec	F-mea	Prec	Rec	F-mea	Prec	Rec	F-mea
1xx	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,94	1,00	0,97	1,00	1,00	1,00	0,85	1,00	0,92
2xx	0,95	0,90	0,92	0,99	0,60	0,75	0,92	0,85	0,88	0,80	0,43	0,56	0,91	0,45	0,60	0,76	0,77	0,76
3xx	0,85	0,82	0,83	0,89	0,67	0,76	0,89	0,79	0,84	0,90	0,71	0,79	0,96	0,42	0,58	0,92	0,50	0,65

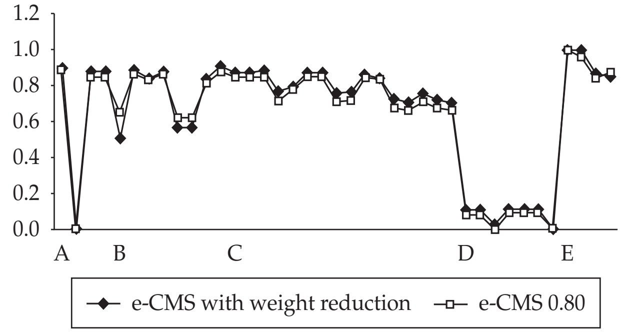


Figure 5. Precision results with original values decreased by 10%.

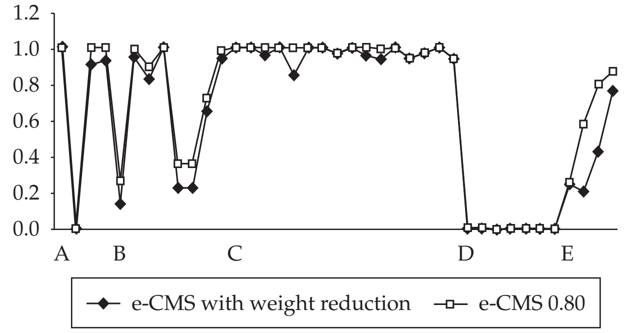


Figure 6. Recall results with original values decreased by 10%.

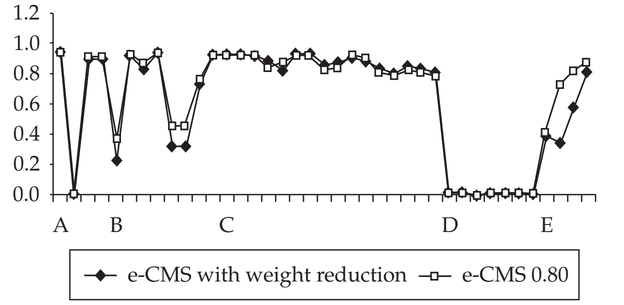


Figure 7. F-Measure results with original values decreased by 10%.

with the addition of *e-CMS* results. It is possible to observe that *e-CMS* showed gains for tests of group 2xx (CMS-MC, DSSim, ONTODNA), whereas for group 3xx, it wins only for CMS-MC.

5. *e-CMS* Applied to Content Generation for Semantic Portals

This section presents the *SiGePoS* System for dynamically generation of content for semantic portals[22], which includes *e-CMS* as an important component of one of its main submodules.

5.1. *SiGePOS*

SiGePoS main goal is to provide an integrated view of information according to domain ontologies, aiming at dynamic organization and publication of content in semantic portals. Figure 8 presents *SiGePoS* architecture. It is composed of four main modules: the search module, responsible for finding regular and semantic Web documents; the semantic and KDD (Knowledge Data Discovery)[11] modules, which are in charge of filtering and organizing the retrieved Web documents; and the instantiation module, responsible for populating the portal, based on the content of the selected Web documents. All these modules interact with the Base Ontology (BO), which is the ontology that supports the portal organization and instantiation. *SiGePoS* modules and their functionalities will be briefly described in this section, in order to illustrate the the *e-CMS* usage.

SiGePoS was developed in Java, combined with the Jena library[VIII] to manipulate ontologies.

The Search module includes a semantic search engine to retrieve semantic documents, i.e., documents that contain ontologies. These selected ontologies are called External Ontologies (EO). These searches are based on query expressions built according to the BO concepts and to the notion of contextualization, mentioned before. Among the semantic search mechanisms available on the Web, we chose the *Swoogle*[5] to implement this module, since it offers the interface REST[10] to execute queries.

The Semantic module is responsible for dynamic instantiation of the BO. It consists of four submodules: *semantic query building*, *filtering*, *matching*, and *mapping*, whose goals are briefly described as follows.

Periodically, according to the system configuration, this module activates the *semantic query building* submodule, the one responsible for formulating and creating queries according to query models, named *templates*. These templates are instantiated into query expressions, which constitute the main entry of the *search module*. These documents are then filtered by the *filtering* submodule, so that only well-formed semantic documents are stored in the system. Once filtered, a selected set of documents, which represent external ontologies (EO), are then considered for the next phase. Then, the *matching* submodule compares BO entities[IX] to the entities of each one of the EOs. *e-CMS* was our choice for the implementation of the matching submodule. From now on, the *mapping* submodule is activated, so that instances of the EO are transferred into the BO mapped concepts.

[VIII] http://jena.sourceforge.net

[IX] In the scope of this paper, entity and concept are used indifferently

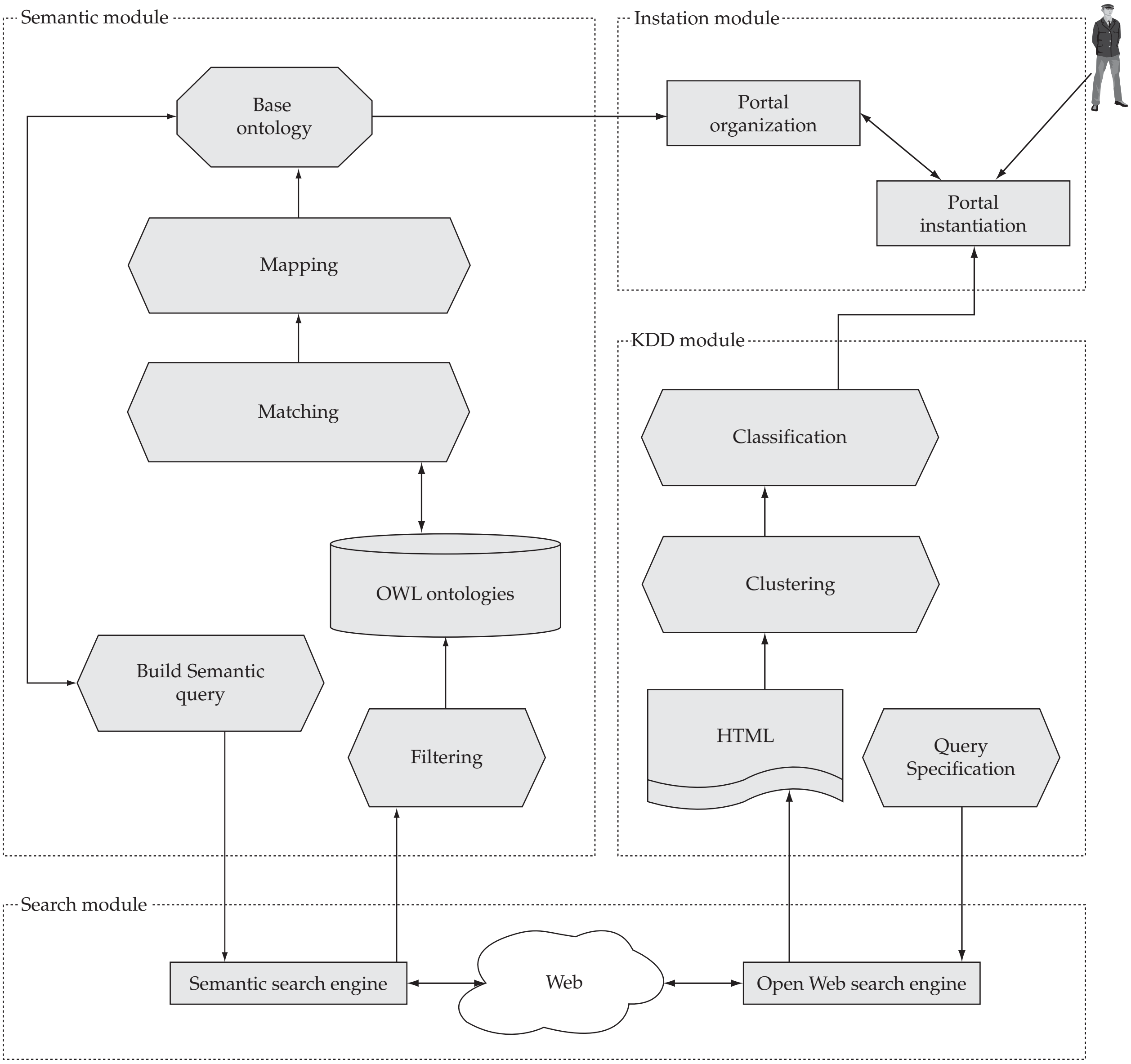


Figure 8. An architecture for Content Generation for Semantic Portals.

The Semantic Query Building submodule is responsible for creating and sending semantic queries to be processed by the search module. It considers the notion of context[32], where an ontology concept may be used in the search expression to restrain the search results.

Based on the domain ontology concepts and properties, we have built a list of terms and corresponding synonyms in Portuguese and in English, which constituted a dictionary on the educational domain. Query expressions are built according to two templates, with the help of this dictionary. Each template is configured based on the following definitions: a) main concepts: correspond to the first level of the ontology, i.e., they are the children of the root concept (thing); b) subordinated concepts: all the concepts that are descendants of the first level; c) equivalent concepts: those concepts that have similarities (synonyms) with the ontology concept; d) domain concepts: those concepts that play the role of subject in the triple: *subject*, *predicate* and *object* of ontological languages; and e) range concepts: those concepts that play the role of *object* in the triple.

Both templates are specified in BNF notation, as described next.

Template 1 (Figure 9) includes main concepts and their subordinated children until the second level of the ontology, where equivalent concepts, taken from the educational dictionary, are associated with each concept in the template.

Template 2 (Figure 10) is defined as a triple, which is composed by the *domain* concept, the *range* concept, and the relationship between them. Additionally, it includes the concepts subordinated to the *range* concept and all the equivalent terms corresponding to all concepts and relationships within the template.

The boolean operator AND is then added to the query, to ensure the relationship and *range* participation in the template.

```
<template_1>::= <Main Concept>
   [<Equivalent Concepts>]
   [<Subordinated Concepts>]
<Main Concept>::= <mainconcept>
<Equivalent Concepts> ::= ε
   | OR <equivalentconcept>
   | OR <Equivalent Concepts>
<Subordinated Concepts>::=
   ε | OR <subordinatedconcept>
    | OR <Subordinated Concepts>
```

Figure 9. BNF specification of Template 1.

```
<template_2>::= <Domain Concept>
   AND <Relationship>
   AND <Range Concept>
   [<Range Subordinated Concepts>]
   [<Equivalent Concepts>]
<Domain Concept>::= <domainconcept>
<Relationship>::= <relationship>
<Range Concept>::= <rangeconcept>
<Subordinated Concepts Range>::= ε
   | OR <subordinatedconcept>
   | OR <Range Subordinated Concepts>
<Equivalent Concepts>::= ε
   | OR <equivalentconcept>
   | OR <Equivalent Concepts>
```

Figure 10. BNF specification of Template 2.

Some query examples expressed using these templates (Figure 11 and Figure 12), will be described further, taking into account our case study on the educational domain, named BOEDU (Base Ontology on EDUcation) and presented in Figure 13. BOEDU was developed as the pillar for the

```
<query_1>::= (Education_Program) OR  (Academic_
Program OR educational_activity OR Course
OR Programa_Acadêmico OR Ensino_acadêmico)
OR  (Higher_Education_Program OR Extensional_
Course_Program)
```

Figure 11. Example of a query instance according to Template 1.

```
<query_2>::= (Finantial_Institution_for_
Research_Support) AND (supplies_Funding) AND
(Academic_Funding) OR (International_Student_
Program_Cooperation OR Scholarships_Support_
Program) OR (Institution_for_Research_Support
OR sponsors OR provides_funding_for OR prove_
recursos_para  OR fomenta  OR Finantial_Aid
OR Finantial_academic_support  OR Finantial_
Assistance  OR Fomento acadêmico  OR Fomento
pesquisa)
```

Figure 12. Example of a query instance according to Template 2.

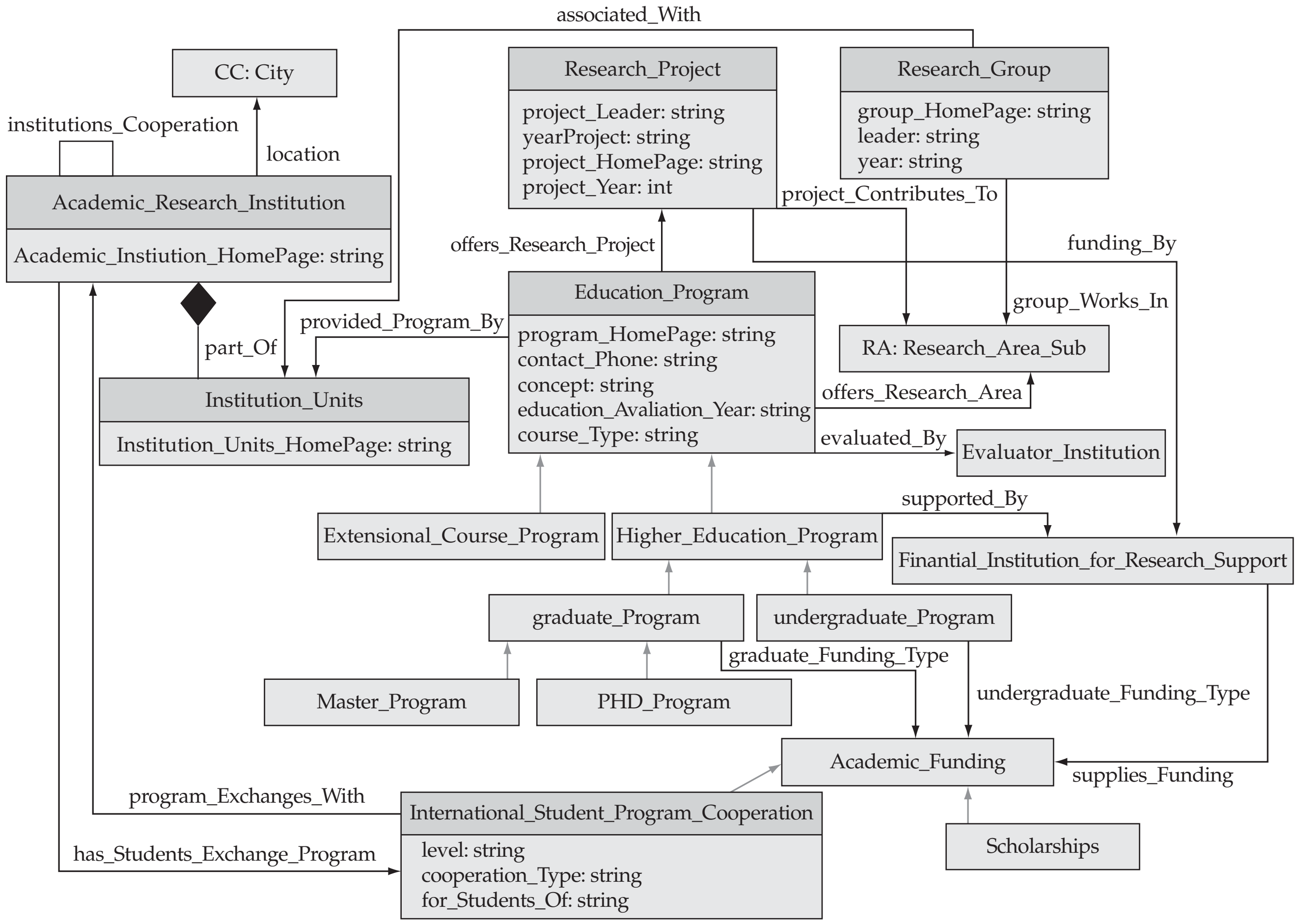


Figure 13. BOEDU – The Base Ontology for the Educational Portal.

semantic portal POSEDU[X] (Semantic Portal in Education)[23], considering concepts of real educational data in Brazil and a list of issues required by a target public with major interest in such an educational portal. However, we also extended our analysis to many educational portals, with a major focus on graduate and undergraduate programs in Brazil and in other countries, as well as on institutional portals that provide grants to support activities such as research and teaching.

The first query example, showed in Figure 11, is based on Template 1. It aims at finding ontologies including higher education programs (Higher_Education_Program) or extensional programs (Extensional_Course_Program).

This query is composed of the main concept *Education_Program* (defining the query context) and its equivalent concepts, both in Portuguese and in English (*Academic_Program, educational_activity, Course , Programa_Acadêmico* and *Ensino_acadêmico*), separated by the Boolean operator OR. Finally, the subordinated concepts *Higher_Education_Program* and *Extensional_Course_Program,* corresponding to the query subject, are associated to the main concept (*Education_Program*).

The second query example presented in Figure 11 is based on Template 2. It aims at finding ontologies corresponding to institutions that provide grants for educational programs, such as graduate and under-graduate programs, offering fellowships and international cooperation programs.

In this example, a triple is composed of: the domain concept Finantial_Institution_for_Research_Support, the relationship *supplies_Funding* and the range concept International_Student_Program_Cooperation and Scholarships_Support_Program, separated by the Boolean operator OR. It considers the synonyms (in both languages) of the domain and subordinated concepts, range and relationships, represented by the terms: *Institution_for_Research_Support*, *sponsors*, *provides_funding_for*, prove_recursos_para, *fomenta*, *Finantial_Aid*, *Finantial_academic_support*, *Finantial_Assistance*, *Fomento acadêmico* and *Fomento pesquisa*.

Hence, the expressions generated based on those templates will be input to the search module, and submitted to semantic and open Web search engines. Based on these expressions, the semantic search engine in *SiGePoS* (*Swoogle*[5]) returns a set of semantic documents, or External Ontologies (EOs).

The set of EOs, returned by the Search module, is then filtered by the Filtering submodule. It discards bad-formed XML documents, and those with erroneous HTTP addresses, selecting only those documents in OWL format. Files with extension DAML and RDF, are also discarded in the current implementation, but should be selected in the near future.

The set of EOs selected by the Filtering submodule becomes available for the Matching submodule. Using *e-CMS*, each EO is matched to the BO. Based on the matching results, EO instances are transferred into the BO entities. This is done by the Mapping submodule.

Finally, although the KDD and Instantiation modules are not directly related to the *e-CMS* mechanism, they both take part in the *SiGePoS* functionality. The KDD module is responsible for filtering, clustering and classifying open Web documents, coming from the Search module. The Instantiation module processes BO instances and a set of related links, respectively coming from the semantic and KDD modules. These instances are used to populate the portal, presenting an organized and updated view of the information on a specific domain (BO domain)[23].

5.2. Case study

In order to demonstrate a real application of the *e-CMS* algorithm, a case study has been developed, considering the ontology of the *Lehigh* University on the educational domain[21]. It is worth observing that this ontology, retrieved from the Web, did present very few instances. Since it represented an interesting educational example for our tests, we decided to increase the number of instances, populating them manually with instances retrieved from its own portal.

Different types of tests have been performed to verify the *e-CMS* matching of the Lehigh university ontology. Similarity calculation results between concepts of the BO and the EO were measured taking into account synonyms of concepts and properties, as well as their influence in the ontologies hierarchy structure.

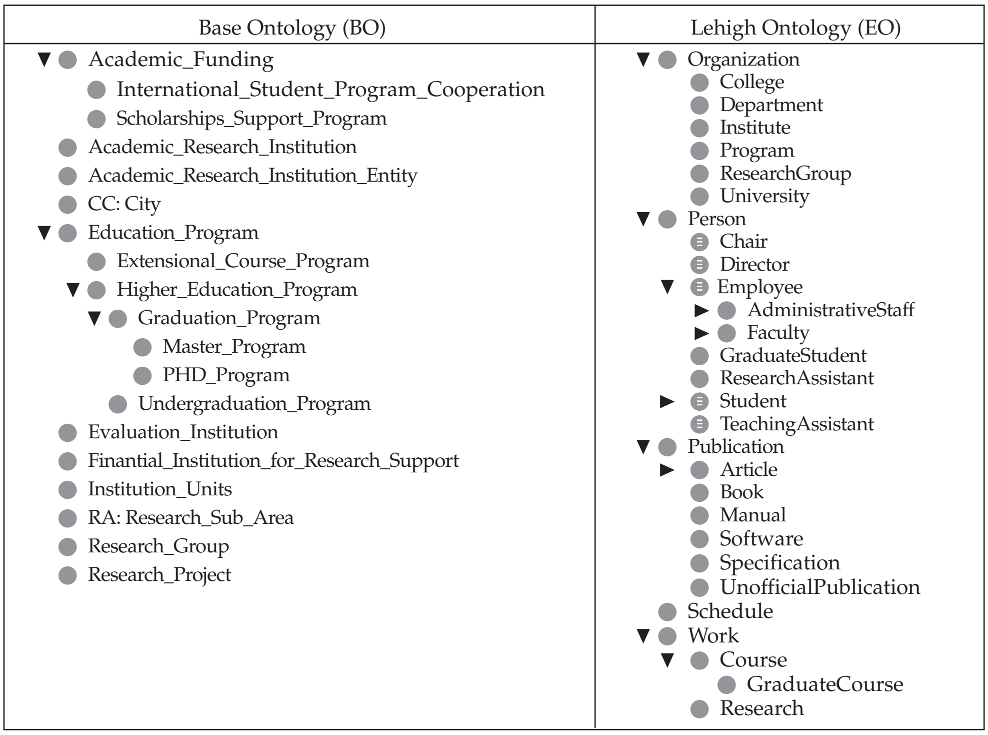
Figure 14 presents the two ontologies that take part in this case study: the BOEDU as the BO, and that of Lehigh University (*NS:http://swat.cse.lehigh.edu/onto/univ-bench.owl\#*), considered as an EO. Table 5 shows some relationships detected in both ontologies.The tests performed are described as follows:

i. With/without using a synonyms dictionary: the first test did not include synonyms. In this case, when compared to the set of matches presented in Table 5, which considered the use of synonyms, only the match between concepts EO:*ResearchGroup* and BO:*Research_Group* have been selected by the *Edit Distance* algorithm, resulting 0.860 as the similarity value. It is worth noticing that even when synonyms are considered, the concepts EO:*Research* BO:*Research_Sub_Area* present a low similarity value (<0.80), meaning this match was discarded. The reason for this is that they did not present any compatible ancestors: *thing* and *work* respectively, which did not have any property in common either.

ii. Modifying hierarchies: the EO hierarchy has been changed in order to evaluate the influence of the concepts hierarchy in both ontologies, as presented in Figure 15. After this modification, the pair of concepts EO:*Research* and BO:*Research_Sub_Area* had a significant increase in its matching similarity value, to 0.898. This was expected, since *Research* and *Research_Sub_Area* were at the same hierarchy level. Similarly, there was an increase from 0.889 to 0.894, on the similarity value of the pair of concepts (EO:University, BO:Academic_Research_Institution).

iii. Eliminating/Creating properties: the EO properties *AffiliatedOrganizationOf, subOrganizationOf,* and *researchProject*, shown in Figure 16, were eliminated.

[X] www.comp.ime.eb.br/~posedu.



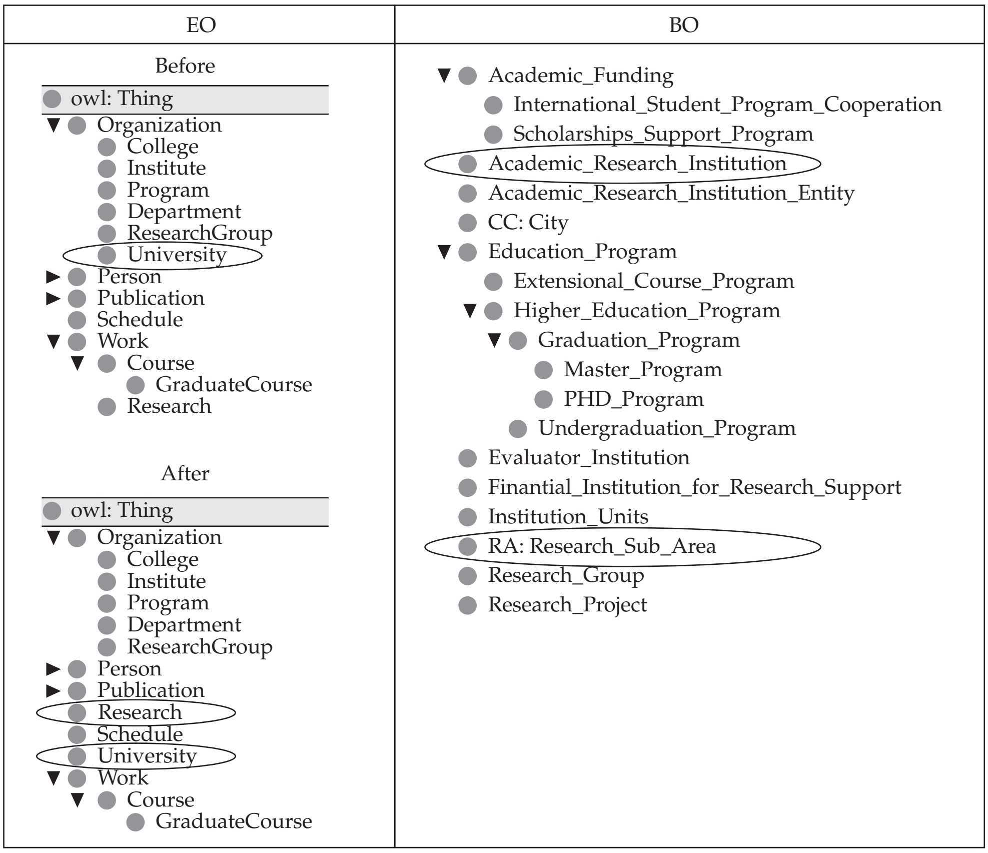
Base Ontology (BO)

- ▼ ● Academic_Funding
  - ● International_Student_Program_Cooperation
  - ● Scholarships_Support_Program
- ● Academic_Research_Institution
- ● Academic_Research_Institution_Entity
- ● CC: City
- ▼ ● Education_Program
  - ● Extensional_Course_Program
  - ▼ ● Higher_Education_Program
    - ▼ ● Graduation_Program
      - ● Master_Program
      - ● PHD_Program
    - ● Undergraduation_Program
- ● Evaluation_Institution
- ● Finantial_Institution_for_Research_Support
- ● Institution_Units
- ● RA: Research_Sub_Area
- ● Research_Group
- ● Research_Project

Lehigh Ontology (EO)

- ▼ ● Organization
  - ● College
  - ● Department
  - ● Institute
  - ● Program
  - ● ResearchGroup
  - ● University
- ▼ ● Person
  - ● Chair
  - ● Director
  - ▼ ● Employee
    - ▶ ● AdministrativeStaff
    - ▶ ● Faculty
  - ● GraduateStudent
  - ● ResearchAssistant
  - ▶ ● Student
  - ● TeachingAssistant
- ▼ ● Publication
  - ▶ ● Article
  - ● Book
  - ● Manual
  - ● Software
  - ● Specification
  - ● UnofficialPublication
- ● Schedule
- ▼ ● Work
  - ▼ ● Course
    - ● GraduateCourse
  - ● Research

Figure 14. Partial view of BO and EO ontologies.



EO

Before

- ● owl: Thing
- ▼ ● Organization
  - ● College
  - ● Institute
  - ● Program
  - ● Department
  - ● ResearchGroup
  - ● University
- ▶ ● Person
- ▶ ● Publication
- ● Schedule
- ▼ ● Work
  - ▼ ● Course
    - ● GraduateCourse
  - ● Research

After

- ● owl: Thing
- ▼ ● Organization
  - ● College
  - ● Institute
  - ● Program
  - ● Department
  - ● ResearchGroup
- ▶ ● Person
- ▶ ● Publication
- ● Research
- ● Schedule
- ● University
- ▼ ● Work
  - ▼ ● Course
    - ● GraduateCourse

BO

- ▼ ● Academic_Funding
  - ● International_Student_Program_Cooperation
  - ● Scholarships_Support_Program
- ● Academic_Research_Institution
- ● Academic_Research_Institution_Entity
- ● CC: City
- ▼ ● Education_Program
  - ● Extensional_Course_Program
  - ▼ ● Higher_Education_Program
    - ▼ ● Graduation_Program
      - ● Master_Program
      - ● PHD_Program
    - ● Undergraduation_Program
- ● Evaluator_Institution
- ● Finantial_Institution_for_Research_Support
- ● Institution_Units
- ● RA: Research_Sub_Area
- ● Research_Group
- ● Research_Project

Figure 15. Changing the hierarchical structure of the EO.

Table 5. Relationships of the EO and BO ontologies.

Properties of the EO			Properties of the BO		
Affiliated Organization_Of	Organization	Organization	Institutions Cooperation	Academic Research Institution	Academic Research Institution
subOrganizationOf	Organization	Organization	Part_Of	Institution Units	Academic Research Institution
Research Project	Research Group	Research	Group Works_In	Research Group	Research Sub_Area

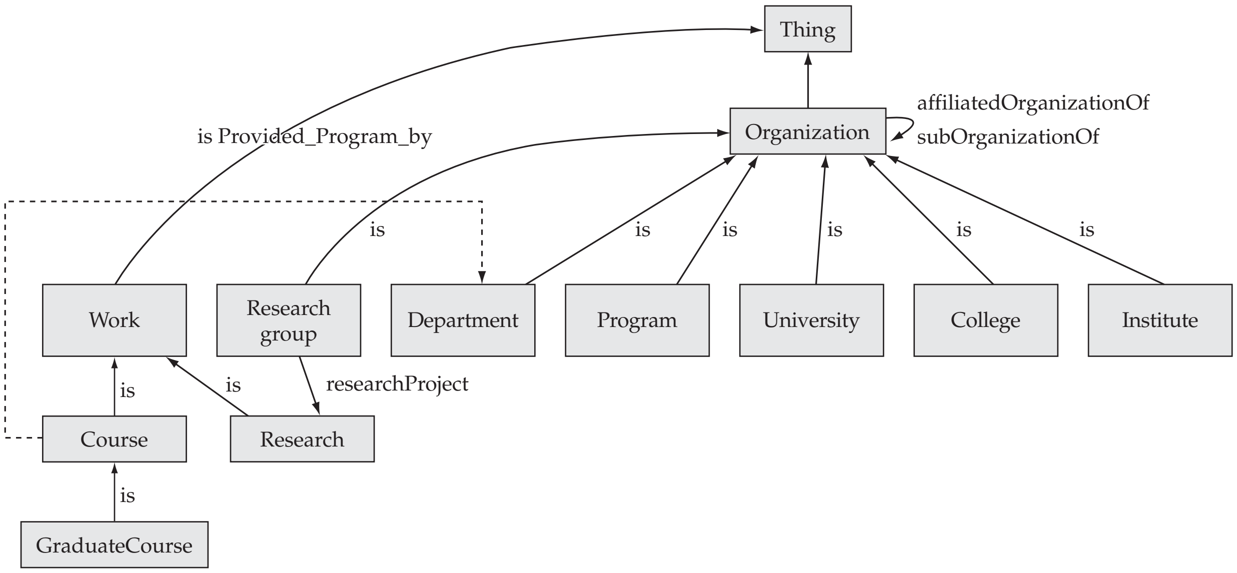


Figure 16. A relationships subset of the EO.

Without this information, similarity values decreased when compared to results shown in Table 6, since only the edit *distance* algorithm and the synonyms dictionary have been considered. The similarity value of the pair of concepts EO:*Organization* and BO:*Academic_Research_Institution* decreased from 0.898 to 0.870. This situation was propagated to both concepts descendants and their corresponding matches. The similarity values of the pair of concepts EO:*ResearchGroup* and BO:*Research_Group* also decreased.

Another interesting situation occurred when property EO:provided_Program_by, was created (as shown in Figure 16), associating the concepts EO:Course and EO:Department. Table 7 shows that similarity values for this matching pair increased from 0.818 to 0.847.

It is worth noticing that whenever instances are transferred, the relationships derived from the corresponding concept properties are also transferred. For example, as an Institution_Unit is part of an Academic_Research_Institution, instances Center_for_Polymer_Science_and_Engineering and Lehigh_University are associated via that property.

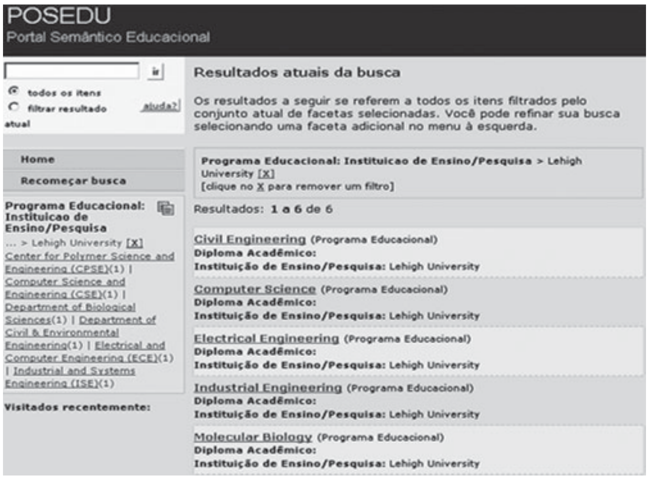
The tests comfirmed it was possible to verify that the *e-CMS* mechanism presented the expected behavior. In order to illustrate the results of the matching and mapping processes for the Lehigh case study, a snapshot of the POSEDU portal is presented in Figure 17, which shows instances of the matched concepts that were transferred to their corresponding concepts at BOEDU. For example, some Educational_Programs (BOEDU) new instances, originally at the Lehigh ontology, such as Civil_Engineering, Computer_Science, Electrical_Engineering, etc., now appear at the POSEDU portal.

Table 6. Matching results considering synonyms.

EO (Lehigh)	BO (BOEDU)	Simil.
Research	Research_Sub_Area	0.716
GraduateCourse	Graduate_Program	0.857
ResearchGroup	Research_Group	0.980
University	Academic_Research_Institution	0.889
Department	Institution_Units	0.887
Course	Education_Program	0.818
Organization	Academic_Research_Institution	0.898
Institute	Institution_Units	0.887

Table 7. Matching results considering properties influence.

EO (Lehigh)	BO (BOEDU)	Similarity
Research	Research Sub Area	0.716
GraduateCourse	Graduation Program	0.857
ResearchGroup	Research Group	0.948
University	Academic Research Institution	0.870
Department	Institution Units	0.876
Course	Education Program	0.847
College	Academic Research Institution	0.870
Organization	Academic Research Institution	0.870
Institute	Institution Units	0.876



Resultados atuais da busca

Os resultados a seguir se referem a todos os itens filtrados pelo conjunto atual de facetas selecionadas. Você pode refinar sua busca selecionando uma faceta adicional no menu à esquerda.

Programa Educacional: Instituicao de Ensino/Pesquisa > Lehigh University [X]
[clique no X para remover um filtro]

Resultados: 1 a 6 de 6

Civil Engineering (Programa Educacional)
Diploma Acadêmico:
Instituição de Ensino/Pesquisa: Lehigh University

Computer Science (Programa Educacional)
Diploma Acadêmico:
Instituição de Ensino/Pesquisa: Lehigh University

Electrical Engineering (Programa Educacional)
Diploma Acadêmico:
Instituição de Ensino/Pesquisa: Lehigh University

Industrial Engineering (Programa Educacional)
Diploma Acadêmico:
Instituição de Ensino/Pesquisa: Lehigh University

Molecular Biology (Programa Educacional)
Diploma Acadêmico:
Instituição de Ensino/Pesquisa: Lehigh University

Programa Educacional: Instituicao de Ensino/Pesquisa
... > Lehigh University [X]
Center for Polymer Science and Engineering (CPSE)(1) | Computer Science and Engineering (CSE)(1) | Department of Biological Sciences(1) | Department of Civil & Environmental Engineering(1) | Electrical and Computer Engineering (ECE)(1) | Industrial and Systems Engineering (ISE)(1)

Figure 17. Adding Lehigh instances to POSEDU**.**

6. Conclusion

This work introduced *e-CMS*, an extension of the *CMS* mechanism for similarity calculation, used in the ontology matching process. *e-CMS* has been evaluated according to the benchmark proposed by the OAEI initiative. Various tests have been applied over a set of ontologies, variating some parameters (threshold and properties weights). They showed that *e-CMS* presented very good results for recall when compared to original *CMS*, and better results, in most cases, for precision and recall when compared to *CMS-StrutCano*. Although *CMS-MC* showed better performance than *e-CMS* on precision, we considered it not very significant, since it was due to the fact *CMS-MC* uses the best results of a pre-defined subset of CMS original algorithm combinations.

e-CMS was developed as part of the *SiGePoS* system (matching module), created to generate content for semantic portals contents within a specific domain.

As future work we intend to increment *e-CMS* with other lexical resources, in order to improve the process of detecting possible redundancies coming from different ontologies, and false agreements. Since the main goal is to allow portal ontology to interoperate with other ontology domains, and considering that the same concept may have different senses, a disambiguation process is required to select the most probable intended sense of a term, considering its possible meanings. Additionally, we intend to explore new proposed algorithms, such as ASMOV and DSSIM, and further incorporate them into future versions of the *SiGePos* system.

Acknowledgements

This work has been supported by FAPERJ (proc. E-26/151.304/2007), CAPES and CNPq (proc. 308809/2005-0).

References

1. Brickley D, Buswell S, Matthews BM, Miller L, Reynolds D, Wilson MD. Semantic web advanced development for Europe (SWAD-Europe). In: *Proceedings of the First International Semantic Web Conference on The Semantic Web*; 2002; Sardinia. p. 409-413.

2. Choi N, Song Il. Y, Han H. Survey on ontology mapping. *ACM SIGMOD Record* 2006; 35(3):34-41.

3. Corcho O, Gómez-Pérez A, López-Cima A, López-Garcia V, Suárez-Figueroa MC. ODESeW: automatic generation of knowledge portals for intranets and extranets. In: *Proceedings of the International Semantic Web*; 2003; Florida. p. 802-817.

4. Dey AK. Understanding and using context. *Personal Ubiquitous Comput*. 2001; 5:4-7.

5. Ding L, Finin T, Joshi A, Pan R, Cost SR, Peng Y. et al. Swoogle: a search and metadata engine for the semantic web. In: *Proceedings of the 13th ACM Conference on Information and Knowledge Management*; 2004; Washington. p. 652-659.

6. Euzenat J, Shvaiko P. Ontology matching. Berlin Heidelberg: Springer-Verlag; 2007.

7. Euzenat J, Valtchev P. Similarity-based ontology alignment in OWL-Lite. In: *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI'04)*; 2004; Valence. p. 333-337.

8. Euzenat J, Isaac A, Meilicke C, Shvaiko P, Stuckenschmidt H, Šváb O. et al. First results of the ontology alignment evaluation initiative 2007. Available from: <http://oaei.ontologymatching.org/2007/results/oaei2007.pdf>. Access in: 12/2008.

9. Felicíssimo CH. *Semantic interoperability on the Web*: a strategy for ontologies taxonomic alignement [Dissertation]. Rio de Janeiro: Pontifícia Universidade Católica - PUC; 2004.

10. Fielding RT. *Architectural styles and the design of network-based software architectures* [Thesis]. Irvine: University of California; 2000.

11. Frawley WJ, Piatetsky-Shapiro G, Matheus CJ. Knowledge discovery in databases: an overview. *AI Magazine*; 1991. 13(3):58-70.

12. Ganter B, Wille R. Applied lattice theory: formal concept analysis. Available from: <http://www.math.tudresden.de/~ganter/psfiles/concept.ps>. Access in: 12/2008.

13. Gruber TR. Towards principles for the design of ontologies used for knowledge sharing. In: Guarino N, Poli R. (Eds.). *Formal ontology in conceptual analysis and knowledge representation*. The Netherlands: Kluwer Academic Publishers; 1993.

14. Hu W, Qu Y. Falcon-AO: a practical ontology matching system. Web Semantics Sci Serv Agents World Wide Web; 2008.

15. Jarrar M, Majer B, Meersman R, Spyns P, Studer R, Sure Y, Volz R. *Web Portal*: complete ontology and portal. Karlsruhe: Institute AIFB, University of Karlsruhe; 2002. Technical Report, Ontoweb Project (Ist-2000-29243).

16. Jean-Mary Y, Kabuka M. Asmov results for OAEI 2007. In: *Proceedings of the 2nd Ontology Matching Workshop*; 2007; Busan.

17. Jean-Mary Y, Kabuka M. ASMOV: results for OAEI 2008. Available from: <http://www.dit.unitn.it/~p2p/OM-2008/oaei08_paper3.pdf>. Access in: 12/2008.

18. Kalfoglou Y, Hu B, Reynolds D, Shadbolt N. Capturing, representing and operationalising semantic integration. Southampton: University of Southampton, School of Electronics and Computer Science; 2005. Advanced Knowledge Technologies Group. Technical Report.

19. Kalfoglou Y, Hu B. CROSI Mapping System (CMS): results of the 2005 Ontology Alignment Contest. Available from: <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-156/paper12.pdf>. Access in: 12/2008.

20. Kiu CC, Lee CS. Ontodna: ontology alignment results for OAEI 2007. In: *Proceedings of the 2nd Ontology Matching Workshop*; 2007; Busan.

21. Lachtim FA. Organization and instantiation of content in semantic portals [Dissertation]. Rio de Janeiro: Instituto de Matemática e Estatística - IME; 2008.

22. Lachtim FA, Moura AMC, Cavalcanti MC. An architecture for dynamic organization and publication in semantic portals. In: *Proceedings of the 10th International Conference on Information Integration and Web-Based Applications and Services (IIWAS) - Emerging Research Projects, Applications and Services Symposium (ERPAS)*; Linz. Linz: ACM Press; 2008. p. 596-599.

23. Lachtim FA, Ferreira G, Gama R, Moura AMC, Cavalcanti MC. POSEDU: a semantic educational portal. In: *Proceedings of the 2nd Brazilian Workshop Semantic Web and Education* (SBIE 2008); 2008; Fortaleza. 1 CD-ROM.

24. Lausen H, Ding Y, Stollberg M, Fensel D, Hernandez R, Han S. Semantic web portals: state-of-the-art survey. *Journal of Knowledge Management* 2005; 9(5):40-49.

25. MacQueen JB. Some methods for classification and analysis of multivariate observations. In: *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*; 1967; Berkeley. p. 281-297.

26. Maedche A, Motik B, Silva N, Volz R. Mafra: a mapping framework for distributed ontologies. In: *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management*; 2002; London. p. 235-250. Ontologies and the Semantic Web.

27. Maedche A, Staab S, Stojanovic N, Studer R, Sure Y. SEmantic portAL: the SEAL approach. In: Creating the Semantic Web. Cambridge: MIT Press; 2001. Available from: http://www.aifb.uni-karlsruhe.de/~sst/Research/Publications/semanticportal.pdf. Access in: 03/2009.

28. Mäkelä E, Hyvönen E, Saarela S, Viljanen K. Ontoviews: a tool for creating semantic web portals. In: *Proceedings of the International Semantic Web Conference*; 2004; Hiroshima. p. 797-811.

29. Mäkelä E, Viljanen K, Lindgren P, Laukkanen M, Hyvönen E. View-based user interfaces for information retrieval on the semantic Web. In: *Proceedings of the ISWC-2005 Workshop End User Semantic Web Interaction*; 2005; Galway.

30. Nagy M, Vargas-Vera M, Motta E. DSSim: managing uncertainty on the semantic Web. In: *Proceedings of the 2nd Ontology Matching Workshop*; 2007; Busan Busan.

31. Nagy M, Vargas-Vera M, Stolarski P. DSSim results for OAEI 2008. Available from: <http://www.dit.unitn.it/~p2p/OM-2008/oaei08_paper5.pdf>. Access in: 12/2008.

32. Pinheiro WA, Moura AMC. An ontology based-approach for semantic search in portals. In: *Proceedings of the WEBSI*; 2004; Saragoza. Saragoza: IEEE Computer Society Press; 2004. p. 127-131.

33. Pinto HS, Gomez-Perez A, Martins JP. Some issues on ontology integration. In: *Proceedings of IJCAI99's Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends*; 1999; Estocolmo. p. 7-12.

34. Ranganathan SR, Gopinath MA. *Prolegomena to library classification*. New York: Asia Publishing House; 1967.

35. Reynolds D, Shabajee P, Cayzer S. Semantic information portals. Available from: <http://portal.acm.org/citation.cfm?id=1013440>. Access in: 08/2008.

36. Silva NAP. *Multi-Dimensional Service-Oriented Ontology Mapping* [Thesis]. Univ. Trás-os-Montes e Alto Douro Vila Real; 2004.

37. Shafer G. A Mathematical Theory of Evidence. Princeton: Princeton University Press; 1976.

38. Suominen O, Viljanen K, Hyvönen E. User-centric faceted search for semantic portals. In: *Proceedings of the European Semantic Web Conference ESWC*; Innsbruck. p. 356-370.

39. Vesanto J, Alhoniemi E. Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks* 2000; 11(3):586-600.