International Journal of the Physical Sciences Vol. 6(17), pp. 4224-4244, 2 September, 2011
Available online at http://www.academicjournals.org/IJPS
ISSN 1992 - 1950 ©2011 Academic Journals

Full Length Research Paper

Ontology mapping using bipartite graph

Aydin SECER[1]*, A. Coskun SONMEZ[2] and Huseyin AYDIN[1]

[1]Department of Mathematics, Faculty of Art and Sciences, Ataturk University, 25000-Erzurum, Turkey.
[2]Faculty of Electrical and Electronics, Yildiz Technical University, Istanbul, Turkey.

Accepted 22 April, 2011

Fast improvement of web technologies have caused a problem of semantic integration between distributed applications. In this respect, sharing and distribution of information are of vital importance for ontologies. Ontologies may be improved by any independent association and still be used by another association. In case an association decides to use an ontology improved by another association, they should make mapping between ontology concepts. Different worldviews assign different meanings to different concepts and defines them differently. Therefore, mapping stands as an inevitable process. Mapping is the job of finding objects that are compatible between two ontologies. Semantic mapping of ontologies by means of bipartite graph matching algorithms has been studied in this paper. A mapping system defined as $O_{source} \xrightarrow{M} O_{t\arg et}$ has been improved. We have named this system BGOM (bipartite graph optimal mapping). BGOM system finds the one-to-one matching between ontologies O_s and O_t which are in similar domains or in same domain. Firstly, two data matrices for ontology concepts O_s and O_t have been obtained. Next, a score matrix has been obtained from general data matrixes by using Levensthein metric. Finally, Kuhn-Munkres optimal assignment algorithm has been used to optimally map the concepts between O_s and O_t . The reason for this is to find one-to-one matches of concepts in the model we have improved. Kuhn-Munkres algorithm is an effective way to find the most similar couples (Tassa, 2007). Consequently, a one-to-one optimal map has been obtained between source and target ontologies. Application, prediction capability and truth values of BGOM system is evaluated by ontology alignment evaluation initiative (OAEI[1]) and satisfactory results have been obtained. Precision, recall and f-measure values of alignment results in the system we have improved, and are compared to the other systems in OAEI campaign and considerably good results have been obtained. Application of BGOM system between source and target ontologies has assisted effectively for solution of ontology mapping problem.

Key words: Ontology, bipartite graph, ontology mapping, Levensthein metric, Kuhn-Munkres optimal assignment algorithm.

INTRODUCTION

WWW is a media which is used as the widest means of information sharing and exchange (Mao and Peng, 2007). It creates semantic diffusion between different applications in WWW, that information heterogeneity, HTML documents and URL addresses are not designed semantically (Mao and Peng, 2007). Therefore, ontologies are key components which are used for formal and open display of data to solve the information heterogeneity problem. Ontologies are useful for many applications like ontology based data access and they provide data which can be processed openly and semantically by machines generally in computer sciences (Rahm and Bernstein,

*Corresponding author. E-mail: aydinsecer@atauni.edu.tr.

2001; Poggi et al., 2008).Semantic web adopts a non-central web agent's architecture to establish a semantic relation between documents (Berners-Lee et al., 2001). In this respect, ontologies are of vital importance for sharing and distribution of information. Actually, it is possible to develop a global ontology carrying the same meaning for all distributed applications, but this is not a considerable method (Bouquet et al., 2004). It is because different communities have different worldviews and they can develop their own ontologies independently. Many ontology mapping methods are developed on that account. Most of them are based on standards of linguistic and structural characteristic similarity (Meilicke and Stuckenschmidt, 2007; Noy and Doan, 2005; Noy and Musen, 2001; Do and Rahm, 2002; Rahm and Bernstein, 2001; Melnik and Garcia-Molina, 2002). Other methods apply to machine learning method to find semantic relations between ontology concepts (Murata, 2003; Berlin and Motro, 2002; Doan et al., 2003). All these methods aim to find one-to-one matching between source and target ontologies. But this operation is not an easy process. Ontologies provide high-level characterization for low data models and an independent interface for information based services. Furthermore, a Web media information of which is marked semantically with ontology provides searching with methods based on semantic explications of question keys which are at higher level than today's searching techniques to find answers to questions. In this respect, ontologies provide presentation of concepts shared in a domain with a set of terms to accelerate the communication between applications and people (Pirró and Talia, 2010).

Ontology based applications should harmonize their own ontologies to achieve semantic integration. This problem is known as ontology alignment (matching) problem. The aim here is to find matches and relations between concepts between different ontologies. Ontology mapping is of vital importance for semantic web because ontology supports various applications such as semantic questioning, re-writing of questions and semantic web service compositions (Pirró and Talia, 2010). Many mapping algorithms are recommended for ontology mapping (Choi et al., 2006). Especially in (Euzenat and Shvaiko, 2007), ontology mapping problem is indicated with comprehensible current solution approaches and a correct definition. As a general thing, today's techniques make use of some research areas such as Bayes decision theory (Tang et al., 2006), information retrieval (Pirró and Talia, 2007) and description logics (Bouquet et al., 2003).

In recent years, evaluation tests have been carried out by international alignment and evaluation initiative (OAEI) to correctness and convenience of the developed algorithms. RDFS, OWL and some random API languages are improved to support the operation of mapping results of today's ontology languages (Euzenat, 2004).

Objective of study

The purpose of this article is to contribute to solving the problem of heterogeneity of information, so a method is developed. The developed method provides one-to-one and optimal mapping of two ontologies explaining the same subject.

Problem statements

Ontology mapping is the process of interrelating information from diverse sources, for example calendars and to do lists, email archives such as physical, psychological and social presence information, documents of all sorts, contacts (including social graphs), search results and advertising and marketing relevance derived from them. In this regard, semantics focuses on the organization of and action upon information by acting as a mediary between heterogeneous data sources which may conflict not only by structure but also context or value (Wikipedia, 2011).

Motivation

First of all, ontologies O_s and O_t are modeled as bipartite graph, and bipartite graph matching algorithm is used for one-to-one matching. Additionally, mapping software practicing this process has been developed. Results of developed BGOM system is indicated in detail with graphics and sheets in the end of the study.

Research questions

Actually a lot of data around the world can be modeled as a graph. By using the methods of graph theory, how these data are optimally matched. To accomplish this, we used a bipartite graph matching algorithm and we obtained very good results.

RELATED WORKS

In this part of the work other systems which are developed for ontology mapping are mentioned.

Agreementmaker consists of a wide range of automatic matching algorithms defined matchers. They have extendable and modular architecture and provide multipurpose user interface, a set of evaluation strategies and manual visual comparison. Besides, they are semi-automatic. Agreementmaker believes that involving the user in the matching process is crucial in finding the mappings that are not found by automatic methods. By taking advantage of the multi-purpose user interface of the agreementmaker, they have been working on a semi-automatic matching approach that ranks concepts according to their relevance and presents to users, the top-k most relevant concepts together with the most likely mappings associated with them. In addition, solution encompasses a feedback loop that extrapolates new correspondences and corrects wrong mappings. One way to further improve it results in the matching track is to incorporate the capability of extending alignments over multiple ontologies, instead of considering only two ontologies at a time (Isabel et al., 2009).

Anchor-flood algorithm consists of two parts. The first one is ontology schema matching Anchor-flood algorithm ranging a set of ontology concepts and properties. Second one is instance matching approach using our Anchor-flood algorithm. This system is used in Java. The main strength of Anchor-flood's schema matching system is the way of minimizing the comparisons between entities, which leads enhancement in running time. In instance matching, this system shows its strength over value and logical transformations. The weak point is the fact that this system ignores some distantly placed aligned pairs in ontology alignment system. In instance matching, it has still, rooms to work in structural transformation (Seddiqui and Aono, 2009).

AROMA divides to three phases: (1) Preprocess phase represents each title with a set of expressions like classes and properties; (2) second phase consists of the occurrence of rules among labels; (3) post-process phase aims to increase the result mapping correctness and to elect unnecessary matches. On anatomy test, AROMA does not use any particular knowledge about biomedical domain. AROMA runs quite fast since it takes benefits of the subsumption relation for pruning the search space. It further optimized the code since last year, and now AROMA needs around 1 min to compute the alignment. This pruning feature used by AROMA partially explained the low recall values obtained last year. For this edition, we enhanced the recall by using also a string equality based matcher before using the lexical similarity based matcher. Since AROMA returns not only equivalence correspondences but also subsumption correspondences, its precision value is negatively influenced. It could be interesting to evaluate results by using semantic precision and recall. The two large directories that were given in previous editions of OAEI are divided into very small sub directories. AROMA cannot align such very small directories because our method is based on a statistical measure and then it needs some large amount of textual data. However, AROMA discovers correspondences when it is applied to the complete directories (David, 2009). ASMOV is automatic ontology matching instrument designed to facilitate mapping of heterogenic data sources which are modeled as ontology. Current ASMOV application shows up matching between concepts, properties and instances including matching between object and data type properties. ASMOV has presented a brief description of an automated alignment tool named ASMOV, analyzed its performance at the 2009 ontology alignment evaluation initiative campaign, and compared it with its 2008 version. The test results show that ASMOV is effective in the ontology alignment realm, and because of its versatility, it performs well in multiple ontology domains such as bibliographic references (benchmark tests) and the biomedical domain (anatomy test). The tests results also showed that ASMOV is a practical tool for real-world applications that require on-the-fly alignments of ontologies (Jean-Mary et al., 2009). DSSim is designed to overcome three difficulties. These are presentation and interpretation problems, quality of semantic web data and effective mapping of large-scale ontologies. DSSim has found that most of the benchmark tests can be used effectively to test various aspects of an ontology mapping system since it provides both real word and generated modified ontologies. The ontologies in the benchmark are conceived in a way that allows anyone to clearly identify system strengths and weaknesses which is an important advantage when future improvements have to be identified. The anatomy, library and mldirectory tests are perfect to verify the additional domain specific or multi lingual domain knowledge. Unfortunately this year, it could not integrate its system with such background knowledge so the results are not as good as they expected (Nagy et al., 2008).

Falcon-AO, which is an important composition of Falcon, is designed as an automatic ontology matching system which will help to provide semantic integration between semantic web applications using different but related ontologies. The proposed matching tasks cover a large portion of real world domains, and the discrepancies between them are significant. Doing experiments on these tasks are helpful to improve algorithms and systems. In order to enhance applicability, they list some warnings as well as their modifications occurring in our experiment procedure, which might aid organizers to correct the problems in the future: (1) the prefix “RDFS” is not bound in “gemetoaei2007.owl” in the environment task and (2) the encoding is inappropriate in the library task, and their modification is replacing “utf-8” by “iso-8859-1” (Wang and Xu, 2009).

Lily consists of interesting and effective matching techniques to find same couples. Lily performs four main functions, such as: generic ontology matching (GOM) method, used for general matching tasks with small-scale ontologies, large-scale ontology matching (LOM) method, used for matching with large-scale ontologies, semantic ontology matching (SOM) method, used to find semantic matching between ontologies, and lily uses web information to find semantic relations with the help of search engines and ontology mapping and debugging are used for better matching results. Strengths for normal size ontologies if they have regular literals or similar structures, lily can achieve satisfactory alignments. Weaknesses lily needs to extract semantic subgraphs for all concepts and properties. It is a time-consuming process. Even though we have improved the efficiency of the extracting algorithm, it still is the bottleneck for the performance of the system (Wang and Xu, 2009).

RiMOM is a framework developed for ontology matching. Different types of mapping strategies may be added to RiROM. Suitable strategies based on properties of input ontology and specified rules are selected as candidates for matching task. Six important steps exist in general matching process of RiROM. Ontology preprocessing and property factor prediction: input ontologies are loaded to memory and ontology graph is established, over and unnecessary information is removed. Then, the ontology property factors which will be used for strategy selection are predicted. Strategy selection: main idea of strategy selection is that if two ontologies have the same property, these strategies based on property information are predominantly selected and if some property factors are very low, these strategies are not selected. When factor meaning label is low, strategy based on character string is used if label correspondence factor, although based on WordNet is not used. Single strategy arrangement: RiMOM uses the selected strategies to find the matches independently. Each strategy reveals a matching result. Matching Combination: in this phase, RiMOM combines the component results obtained by selected strategies. This combination is carried out with linear interpolation method. Correspondence diffusion (Optional): if two ontologies have high structure correspondence factor, RiMOM uses correspondence propagation process to find new components according to structural information and refine these components. Matching refinement: this refines the components that emerged in the previous step. Several heuristic rules which will eliminate the “non-reliable” matches are also defined in RiMOM.

SAMBO and SAMBO dtf is based upon a framework. This framework consists of two parts. First part calculates the matching recommendations. Second part interacts with user about deciding the last matching. A matching algorithm takes two source ontologies as input. Algorithm includes one or more matchers, and these calculate the correspondence values between expressions coming from different source ontologies. Matchers can use information from a different source. Matching suggestions are determined with composition and filtration composed by one or more matchers. Different combination strategies are acquired by using combination and filtration of results in different ways. Suggestions are provided to the users who will accept or reject these. Acceptance or rejection of suggestion affects the next suggestion. Additionally, failure controller is used to avoid the failures due to combination relations. Output of mapping algorithms is a set of matching relations among expressions coming from source ontologies. A problem that users face is that often it is not clear how to get the best alignment results given that there are many strategies to choose from. In most systems there is usually no strategy for choosing the matchers, combinations and filters in an optimal way. Therefore, they used their experience from previous evaluations to decide which matchers and thresholds to use for which task. The lack of an optimization strategy is also the reason why they did not provide results for the second and third test for anatomy (optimization with respect to precision and recall, respectively). In the future, however, this may be possible using suggestion methods for alignment strategies as proposed, in that they will be able to recommend matchers, combinations and filters based on the alignment task and evaluation methods (Lambrix et al., 2008). SOBOM is an automatic ontology matching instrument. It has three matcher applied in the current version: Linguistic matcher I-Sub, Structural matcher SISF (semantic inductive similarity flooding), which is inspired from Anchor-Promt and SF algorithms and Realtion matcher R-matcher, which makes use of SISF results to acquire matching relations. Furthermore, an ontology former is combined with SOBOM to extract sub-ontologies according to I-Sub results. SOBOM method is totally sequential, therefore does not take into account how to combine the results of different matchers. Strengths: SOBOM deals with ontology from two different views and combines results of every step in sequential way. If the ontologies have regular literals and hierarchical structures, SOBOM can achieve satisfactory alignments. And it can avoid missing alignment in many block matching methods. Weaknesses: SOBOM needs the anchor concepts to extract sub-ontologies. So it heavily depends on the anchor concepts. If the literals of concept missed, SOBOM will get bad results (Xu et al., 2009).

TaxoMap is a mapping instrument which aims to find high compliance between concepts. It applies a centralized mapping, (from a source to target ontology) and takes labels and sub-class definitions into account. This new definition of TaxoMap particularly reduces the working time and provides a parametric structure by determining ontology language and using different threshold values to reveal different mapping relations.

The following improvements can be made to obtain better results:

1. To take into account all concepts properties instead of only the hierarchical ones.

2. Use of WordNet as a dictionary of synonymy. The synsets can enrich the terminological alignment process if a-priori disambiguation is made.

3. To develop the remaining structural techniques which proved to be efficient in the last experiments (Hamdi et al., 2009).PRIOR+ is developed from PRIOR. In addition to character string metric (edit distance) and

profile correspondence of element names used in PRIOR, PRIOR+ takes the structural correspondence into account and adaptively combines different correspondences based on compliances of these correspondences. Moreover, there is constraint satisfaction resolution based on a new artificial neural network in PRIOR+. Parameter tuning is an important issue in the implementation of neural network in PRIOR future work. Another possible improvement is to integrate auxiliary information and web information for ontology mapping. For example, auxiliary information such as WordNet can be used to process synonyms. The co-occurrence of two elements returned by search engines can contribute to identify their semantic relation (Mao and Peng, 2007).

X-SOM is designed to automatically find useful relations between ontological presentations to achieve ontology-based data combination and addition. Theoretic framework used in X-SOM are DL ontology frameworks, but X-SOM is a very flexible approach and it may be considered that it can be expanded to other ontology languages and even other data models like XML and related models. X-SOM is planning to introduce new modules able to extract and reuse the consensual knowledge that emerges in collaborative and social web-applications, in order to disambiguate some mapping situations that generally need user intervention. X-SOM is currently exploring other machine-learning techniques for the matchings combination task, in particular white-box techniques like decision-tree learning. At the moment, the matching strategy is determined by the user; it aim at introducing techniques to suggest a suitable strategy using a-priori analysis of the input ontologies, and make it adaptive during the matching process (Curino et al., 2007). UFOme is a framework of ontology mapping software regulated and applied to help the users about designing and using extensive mapping systems. It is based upon a library of functions of implementing mapping models such as exploring mappings and evaluating mapping strategies. Especially strategy prediction module of designed framework can “predict” the mapping modules used and parameter values (such as weights and thresholds). As future work, in order to provide a solid support to the usability claims of UFOme, a comprehensive usability study has to be carried out. To cope with this, standard usability tests involving real users have to be performed. As a viable methodology to evaluate usability, the SUMI questionnaire can be adopted. Besides, they plan to release a beta version of UFOme in order to collect useful suggestions from real user of ontology mapping systems. It would be also useful to release specific guidelines in order to allow developers to design and implement mapping models fulfilling the architectural requirements of the UFOme underlying software architecture. This way the system can encompass several components and hopefully be largely adopted in the context of ontology mapping (Pirró and Talia, 2010).

DEFINITIONS

Pre-definitions which will be used in the paper will be indicated in this part.

Ontology

It is possible to define ontology with a sextet in the form:

$$O = \left\langle C, P, H^c, H^p, A, I \right\rangle \qquad (1)$$

Here, *C* concepts and *P* properties, give the hierarchical system according to the relations of H^c concepts H^p properties. *A* indicates a set of axioms and *I* indicates the samples of concepts and properties. Standard languages like RDF and OWL are used to specify the hierarchies of classes and properties. For instance, H^c owl: class and rdfs: subclassof are represented with RDFS: property and RDFS: subpropertyof notations in H^p (Pirró and Talia, 2010).Matrix form of Figure 1 is as follows. This matrix will be *G* general data matrix which we will define later. This matrix is calculated separately for both source and target ontologies and this will be indicated in detail later. If we define the earlier ontology as source ontology, our general data matrix is;

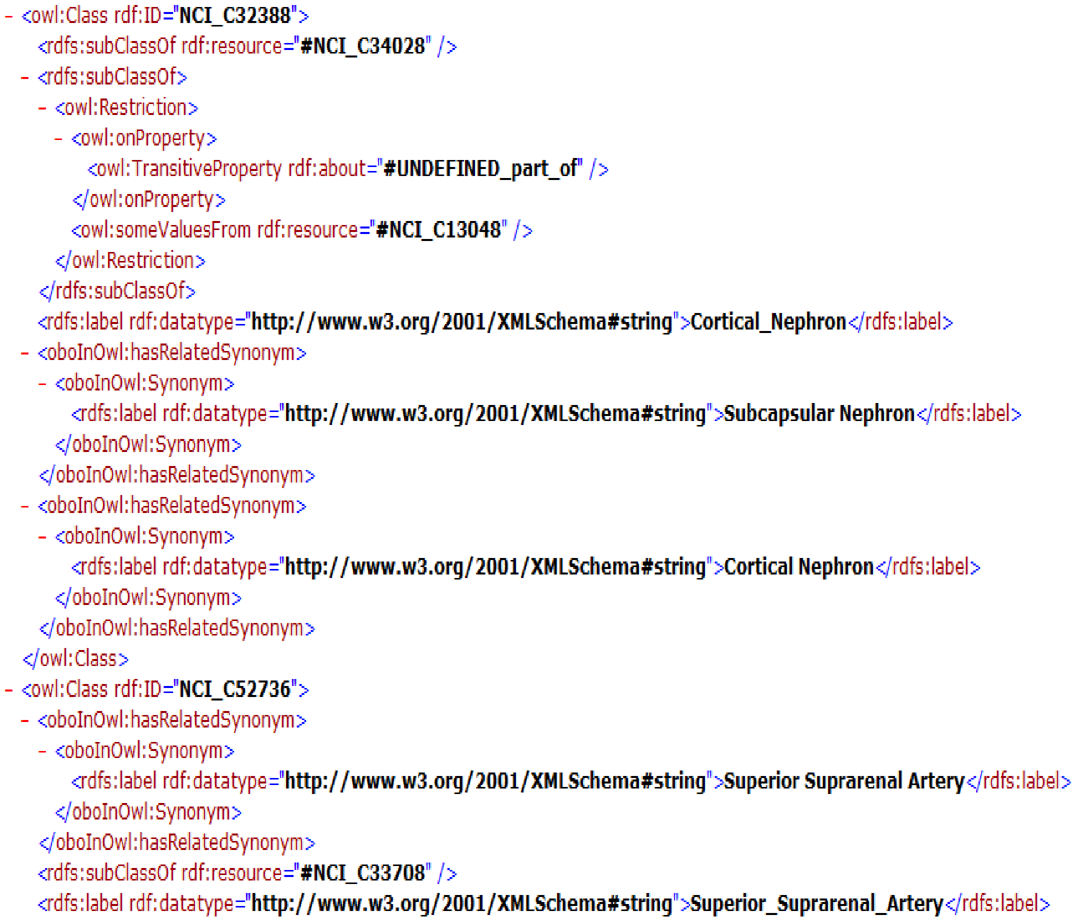
$$G_s = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mn} \end{pmatrix} =$$

$$\begin{pmatrix} Cortical\_Nephron & \{SubcapsularNephron, CorticalNephron\} & null & null & NCI\_C34028 & NCI\_C13048 & null \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} & p_{26} & p_{27}] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{m1} & p_{m2} & p_{m3} & p_{m4} & p_{m5} & p_{m6} & p_{m7} \end{pmatrix} \qquad (2)$$

and the general data can be defined as matrix for target ontology in similar way.

Presentation of mapping

Although ontology mapping problem attracts a great deal of attention from scientific community, still there is no standardized format to stock the ontology mappings (Pirró and Talia, 2010). Ontology languages like OWL provides situated structures for presentation of equivalence between concepts (such as OWL: equivalentClass),



```
- <owl:Class rdf:ID="NCI_C32388">
    <rdfs:subClassOf rdf:resource="#NCI_C34028" />
  - <rdfs:subClassOf>
    - <owl:Restriction>
      - <owl:onProperty>
          <owl:TransitiveProperty rdf:about="#UNDEFINED_part_of" />
        </owl:onProperty>
        <owl:someValuesFrom rdf:resource="#NCI_C13048" />
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Cortical_Nephron</rdfs:label>
  - <oboInOwl:hasRelatedSynonym>
    - <oboInOwl:Synonym>
        <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Subcapsular Nephron</rdfs:label>
      </oboInOwl:Synonym>
    </oboInOwl:hasRelatedSynonym>
  - <oboInOwl:hasRelatedSynonym>
    - <oboInOwl:Synonym>
        <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Cortical Nephron</rdfs:label>
      </oboInOwl:Synonym>
    </oboInOwl:hasRelatedSynonym>
  </owl:Class>
- <owl:Class rdf:ID="NCI_C52736">
  - <oboInOwl:hasRelatedSynonym>
    - <oboInOwl:Synonym>
        <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Superior Suprarenal Artery</rdfs:label>
      </oboInOwl:Synonym>
    </oboInOwl:hasRelatedSynonym>
    <rdfs:subClassOf rdf:resource="#NCI_C33708" />
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Superior_Suprarenal_Artery</rdfs:label>
```

Figure 1. A part of OWL ontology in RDF format.

relations (OWL: equivalentProperty) and instances (owl: SameAs). This approach permits OWL extraction engines to automatically interpret the meanings of mapping and question the different ontologies (Pirró and Talia, 2010). But a reliable value (interval) cannot be interpreted. Furthermore, presentation defined in (Euzenat, 2004) is adopted because it provides several classes of flexible and applicable semantic applications. We have given emphasis to explore one-to-one mapping. That is, we have aimed to find the most correspondent existence in O_t for each existence in O_s.

Ontology mapping

An ontology mapping M is defined with the quartet form below:

$$O_s \xrightarrow{M} O_t \Rightarrow M := \langle o_s, o_t, r, k \rangle \tag{3}$$

In this equation, o_s is a concept of O_s ; o_t is a concept of O_t , r is a type of (=, <, >, <=, >=, <>) relation and similarity measurement between $k \to [0,1]$, o_s and o_t concepts. O_s and O_t are two different ontologies defining the same source. Ontology mapping is creation of an M function which finds real world object pairs (o_s, o_t) when $o_s \in O_s$ and $o_t \in O_t$. In fact, mapping is developing a convention between two sets or in other words a system. This system is a mechanism which finds the most correspondent objects between two sets. M function here as corresponds to a system. For instance, in this article, M function is the epitome of the method that we have developed. Figure 2 earlier represents a map of M = Map matching two ontologies obtained from two different database. Actually, ontology mapping can be summarized with this shape.

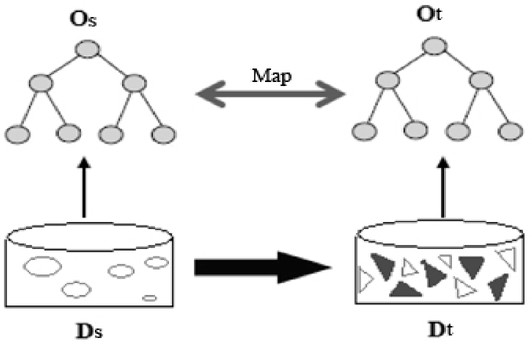


Figure 2. Ontology mapping problem.

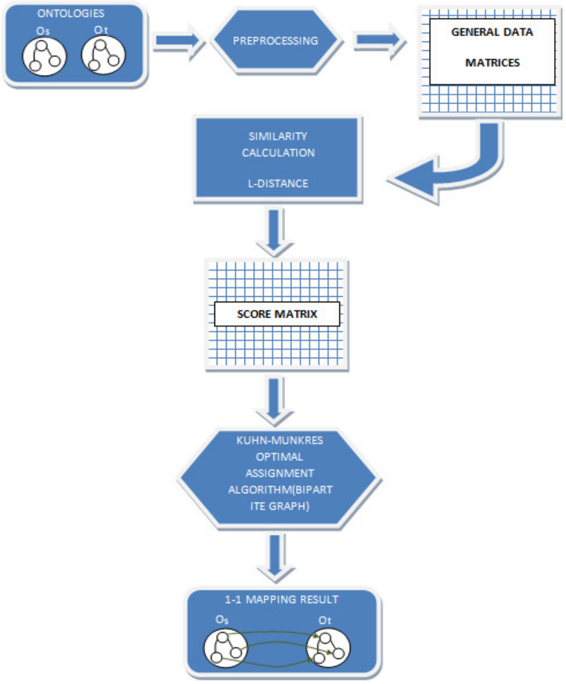


Figure 3. Bipartite graph based matching model (BGOM architecture).

Bipartite graph based approach model (BGOM)

Information heterogeneity in WWW poses an obstacle for semantic integration (Mao and Peng, 2007). Though different techniques have been examined to find mappings, little work is made to solve constraint satisfaction problem for ontology mapping (Mao, 2008).

Many methods to provide semantic integration are available and still being developed. The purpose of this study is to solve the ontology mapping problem and make it possible to establish a semantic relation between heterogenic data. We named out method as BGOM (bipartite graph based optimal model). BGOM aims to find semantic correspondence between ontology concepts and provide more accessible heterogenic data by using ontology mapping. In this study, elements define ontology concepts or classes. BGOM architecture is based on weighted bipartite graph and so we should calculate the edge weights firstly. There are many methods of character string matching in the matter of calculating edge weights. Here, every graph node will keep score of an ontology concept and every edge weight will keep score of correspondence of two concepts. In this study, we used Levensthein metric (edit distance) to calculate concept correspondence. Following figure shows BGOM architecture.

As shown in Figure 3 earlier, BGOM architecture takes two ontologies in RDF format as input. One of these ontologies is source ontology and the other is target ontology. Firstly, these ontologies are preprocessed. Secondly, classId, labels, synonyms, sub and super classes, abbreviations and definitions are obtained. Then, a general data matrix is formed for each ontology. After this process, labels and equivalents of two ontologies are combined with Levensthein metric and thus we obtain the score matrix. Synonyms have been considered as labels, in other words as properties. Indeed, edit distance have been used here. That is, correspondences have not been measured with a semantic approach. Two concepts have been compared with label first and then with equivalent and score of the one which is more correspondent has been kept. In the last process, we give this score matrix to Kuhn-Munkres bipartite graph optimal matching algorithm as input and carry on the iteration until the optimal matching finishes. As a consequence, we obtain an optimal matching map between source and target ontology concepts. Now, we will introduce BGOM steps in detail.

Preprocessing ontologies

Firstly, input ontologies should be preprocessed. We have developed an RDF ontology parser in .Net platform to achieve this. This parser extracts all elements like labels, equivalents, sub and super classes, definitions and classid from ontologies. This process is a pre-preparation to establish a general data matrix.

Establishment of general data matrix

After ontologies which are given in RDF format are preprocessed, a general data matrix, the columns of which is composed of label, equivalent, sub and super class, definition and classId elements. Each line of matrix represents a concept in ontologies. Elements in lines keep relations with sub and super classes and hierarchical information. If O_s has *m* components and *n* concepts, then we define the general data matrix of O_s as follows:

$$G_s = \begin{pmatrix} p11 & p12 & ... & p1n \\ p21 & p22 & ... & p2n \\ : & : & : & : \\ pm1 & pm2 & ... & pmn \end{pmatrix}_{m\times n} \qquad (4)$$

Each line of G_s defines a concept of O_s and each column component defines concept components. These specify labels, equivalents, properties, hierarchical information and constraint. Some column values can be blank, because some concepts may not have this component. Similarly, G_t general data matrix of O_t can be defined as follows:

$$G_t = \begin{pmatrix} p11 & p12 & ... & p1n \\ p21 & p22 & ... & p2n \\ : & : & : & : \\ pk1 & pk2 & ... & pkn \end{pmatrix}_{k\times n} \qquad (5)$$

Levensthein metric and score matrix

The edit-distance *d(x, y*) of two strings *x* and *y* is the minimal cost of a sequence of symbols insertions, deletions or substitutions transforming one string into the other:

$$d(x,y) = \min_{h(w)=(x,y)} c(w) \qquad (6)$$

When *c* is the function defined by $c(a,a)=0$ and $c(a,\varepsilon)=c(\varepsilon,a)=c(a,b)=1$ for all *a, b* in *Σ* such that $a \neq b$, the edit-distance is also known as the Levensthein distance (Allauzen and Mohri, 2009).

L-Distance is one of the methods which are mostly used in the calculation of conceptual correspondence. This part is totally optional for our system. This method is only used to find score matrix, but all other different methods can also be used. Our aim is to apply the best method for one-to-one matching of this matrix after obtaining score matrix (Figure 4). We have achieved this with Kuhn-Munker algorithm. In computer sciences and informatics, Levensthein metric is a metric which measures the amount of difference between two character strings. Levensthein distance between two character strings is established with transformation of a character in character string to another character string via addition, deletion and displacement.

Metric have taken this name in 1965 after Vladamir Levensthein who thought of this metric. Generally, it is used in applications where it is needed to determine how different or similar two different character strings are, like syllable controller. It can also be thought as an extension of Hamming metric which is used for character strings of same length and which uses only displacement arrangement strategy. Levensthein metric uses a matrix of size (*m* + 1) × (*n* + 1) for calculation of correspondence between two character string whose lengths are m and n, respectively. This matrix is defined as follows:

$$L = \begin{pmatrix} L_{11} & L_{12} & ... & L_{1n+1} \\ L_{21} & L_{22} & ... & L_{2n+1} \\ : & : & : & : \\ L_{m1} & L_{m2} & ... & L_{m+1n+1} \end{pmatrix}_{(m+1)\times(n+1)} \qquad (7)$$

Pseudo code

```
proc editDist(A, B,Score)
n   length[A]
m   length[B]
Edit[0, 0]   0
for i   1 to n do
Edit[i, 0]   Edit[i – 1, 0]+ Score [S[i], _]
   for j   1 to n do
Edit[0, j]   Edit[0, j – 1]+ Score[_, T[j]]
for i   1 to n do
for j   1 to m do
                    Op1   Edit[i – 1, j]+ Score [S[i], _]
Op2   Edit[i, j – 1]+ Score [_, T[j]]
Op3   Edit[i – 1, j – 1]+ Score [S[i], T[j]]
Edit[i, j] = min{Op1,Op2,Op3}
return Edit[n,m]
```

BGOM method that we developed uses this matrix to form score matrix. With the help of this matrix, correspondence of each concept of two ontologies is calculated as Cartesian which is a score value and written in a matrix form. For instance, we discuss $o_{11} \in O_1$ and $o_{21} \in O_2$ ontology concepts. If o_{11} and o_{21} are equivalent objects, score value is $L[o_{11}o_{21}] = 0$. If o_{11} and o_{21} are not equivalent, score value is $0 < L[o_{11}o_{21}] < e$, $e \in N$. *e* is a natural number here and varies according to lengths and correspondences of two character string. If O_1 has *m* concepts and O_2 has *n* concepts, score matrix the matrix of size which is *m x n.* Hereunder, we can define score matrix as follows

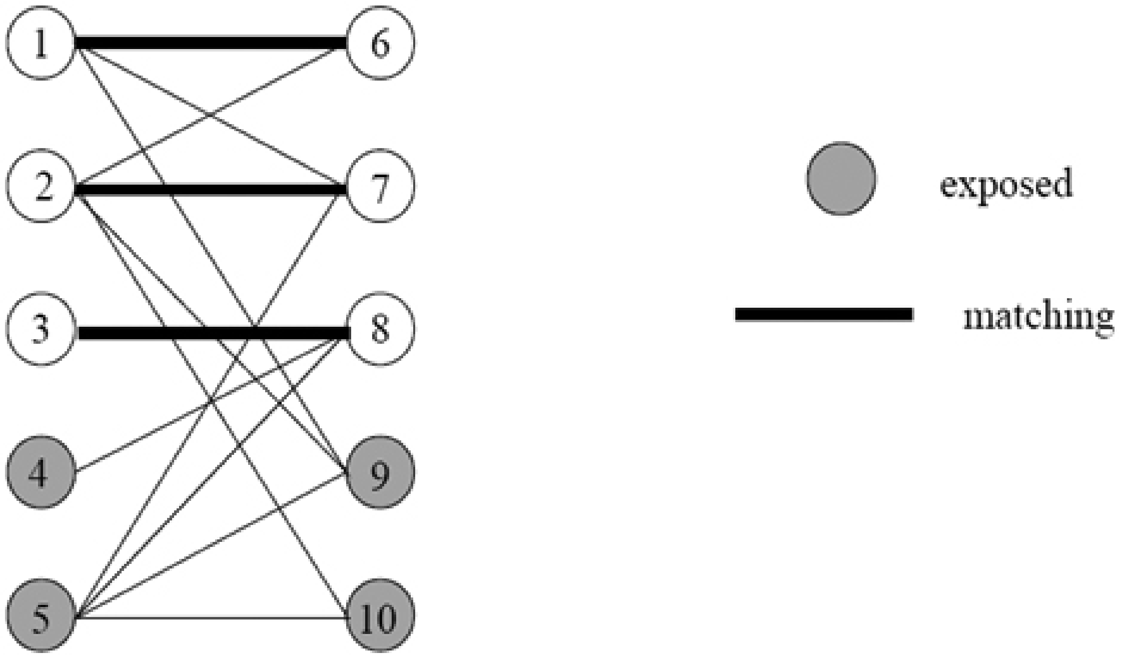


Figure 4. While (1; 6), (2; 7) and (3; 8) edges match one-to-one, 4, 5, 9 and 10 points remain open.

$$S=\begin{pmatrix} L[o_{11}o_{21}] & L[o_{12}o_{21}] & L[o_{13}o_{21}] & \dots & L[o_{1m}o_{21}] \\ L[o_{11}o_{22}] & L[o_{12}o_{22}] & L[o_{13}o_{22}] & \dots & L[o_{1m}o_{22}] \\ L[o_{11}o_{23}] & L[o_{12}o_{23}] & L[o_{13}o_{23}] & \dots & L[o_{1m}o_{23}] \\ : & : & : & \dots & : \\ L[o_{11}o_{2n}] & L[o_{12}o_{2n}] & L[o_{13}o_{2n}] & \dots & L[o_{1m}o_{2n}] \end{pmatrix}_{m\times n} \qquad (8)$$

After score matrix is formed, we will consider this matrix as bipartite graph, edge weights of which is clear, for optimal mapping of ontology concepts.

Bipartite graph matching

When *V* is a set of vertex and *E* is a set of edges, a *G* graph is defined as

$$G=(V;E) \qquad (9)$$

In a $e=(u;v)$ edge, *u* and *v* are end points and adjacent to *e, u* and *v.* Essential condition for $G=(V;E)$ graph to be bipartite is that it is composed of two discrete sets like *A* and *B* and no point in discrete sets matches with a point in the same set. Let's think of a matching such as $M \subseteq E$. If no point remains open in matching, or in other words if matching cardinality is $|A|=|B|$, matching is perfect.

Bipartite graph modeling of source and target ontologies

A bipartite graph is a graph where points are divided into two discrete sets and where two points in the same set do not match. Many data in real world can be modeled as bipartite graph. These models can cover terms, documents, texts, customers and market products. We have considered ontologies as bipartite graph. This graph has two different node sets. One of these sets is classes of source ontologies and the other is classes of target ontologies. Each edge of graph keeps weight scores of correspondence of relations between concepts. Each class in source ontology is connected to the class in target ontology with relation edges. In compliance with the nature of bipartite graph, no concept includes a relation score with the concept in the same ontology. As we mentioned above, it defines the relation between two concepts. We can use a threshold value that we have determined earlier to fasten the algorithm we developed and to provide more explicit results. By this way, we avoid unnecessary matching. Many algorithms like Levensthein metric which is one of today's character string correspondence algorithms in calculating score values (Rodríguez and Egenhofer, 2003).

Minimum weighted optimal assignment problem

Score matrix is calculated after two ontologies are formed

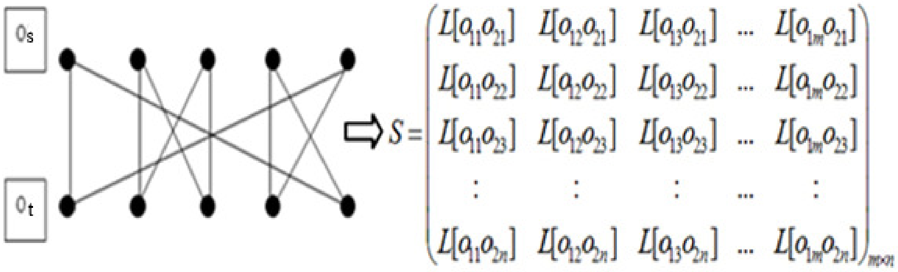

0s
0t
S =
L[o11o21] L[o12o21] L[o13o21] ... L[o1mo21]
L[o11o22] L[o12o22] L[o13o22] ... L[o1mo22]
L[o11o23] L[o12o23] L[o13o23] ... L[o1mo23]
L[o11o2n] L[o12o2n] L[o13o2n] ... L[o1mo2n]
m×n

Figure 5. Score matrix.

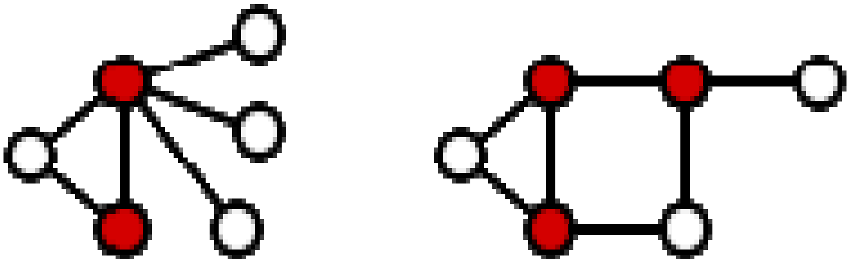


Figure 6. Vertex cover.

as bipartite graph. If $i, j \in E$, minimum weighted score matrix $S = S(i, j)$ is calculated. This matrix represents mutual correspondence distance of every class. Figure 5 shows the transformation of bipartite graph into score matrix.

Vertex cover

Formally, coverage of vertexes in a G graph means the set of points where each edge of *G* is adjacent to at least one of the elements in *C*. Red nodes in figure 6 represent the vertex covers.

Theorem 1 (König)

In a bipartite G graph, matching of maximum number is equal to coverage of vertexes of minimum number.

KUHN-MUNKRES ALGORITHM (HUNGARIAN METHOD)

Let *A* be a $n \times n$ matrix. The following algorithm finds a permutation $\pi \in S_n$ that minimizes the expression $\sum_i A_{i,\pi(i)}$. In this algorithm, the entries of the matrix *A* are being modified repeatedly. Zero entries in the modified matrix may be either marked, by a star or by a prime, or unmarked. In addition, each row or column in the matrix may be either covered or uncovered. Initially, there are no starred or primed entries in the matrix and none of the rows or columns is covered (Tassa, 2007).

1. For each row in the matrix *A* find its minimal entry and subtract it from all entries in that row.

2. For all $1 \le i$, $j \le n$ if $A_{i,j} = 0$ then star that zero entry, unless there is already a starred zero in the same row or in the same column.

3. Cover each column that contains a starred zero. If all columns are covered, go to Step 7.

4. Repeat the following procedure until there are no uncovered zeros left, and then go to Step 6: find an uncovered zero and prime it. If there are no starred zeros in the same row as this primed zero, go to Step 5. Otherwise, cover this row and uncover the column containing the starred zero.

5. Construct a series of alternating primed and starred zeros as follows. Let z_0 be the uncovered primed zero that was found in Step 4. Let *z*1 be the starred zero in the column of z_1 (if any). Let z_2 be the primed zero in the row of z_1 (there will always be one). Continue to construct this series of alternating primed and starred zeros until it terminates with a primed zero that has no starred zero in its column. Unstar each starred zero of the series, star each primed zero of the series, erase all primes and uncover all rows and columns in the matrix. Go to Step 3.

6. Find the smallest uncovered value, add it to every entry in each covered row, and subtract it from every entry in each uncovered column. Go to Step 4.

7. At this stage, in each row of the matrix, as well as in each column, there is exactly one starred zero. The positions of the

starred zeros describe an optimal permutation $\pi \in S_n$. Output this permutation and stop. We think a bipartite *G* graph parts of which are O_s and O_t. When *w* is weight (score) measure, $\forall e \in G$ edge is given as $w(o_{si}, o_{tj})$. The problem here is to find an assignment which has minimum weight. One of the solutions of this problem is Kuhn-Munkres method. We focus to find a way in *D* graph derived from *G* in every step of Kuhn-Munkres algorithm. If we begin from the start line and choose a minimum weighted way in each step, we find an assignment which has minimum cost (if points of an edge in D forms an edge which connects O_t to O_s, its weight should be .... with -1). We think $|O_s|=|O_t|=n$ here. In this respect, method can be indicated with the sample below. When $O_s = \{o_{s1}, o_{s2}, o_{s3}, o_{s4}\}$ and $O_t = \{o_{t1}, o_{t2}, o_{t3}, o_{t4}\}$ and score matrix below is given:

$$\begin{array}{c|cccc} & o_{t1} & o_{t2} & o_{t3} & o_{t4} \\ \hline o_{s1} & 8 & 16 & 14 & 20 \\ o_{s2} & 7 & 7 & 8 & 8 \\ o_{s3} & 9 & 21 & 21 & 23 \\ o_{s4} & 7 & 19 & 15 & 20 \end{array} \tag{10}$$

At the first glance, assignment on matrix consists of *n=4* inputs for each line and each column. Value of assignment is total of selected inputs. Thus, if we add a δ constant to or remove δ constant from an input, value of assignment increases or decreases in amount of δ. Therefore, value of optimal assignment always remains optimal and its value decreases in amount of δ. It also applies to columns. So, we obtain a matrix which has zero inputs in each line and which does not have negative inputs via taking the minimum value difference of each line. We do the same process for columns and we maintain the process till we obtain a matrix which has at least a zero input and which is not negative. The matrix below is obtained after all processes.

$$\begin{array}{c|cccc} & o_{t1} & o_{t2} & o_{t3} & o_{t4} \\ \hline o_{s1} & 0 & 8 & 5 & 11 \\ o_{s2} & 0 & 0 & 0 & 0 \\ o_{s3} & 0 & 12 & 11 & 13 \\ o_{s4} & 0 & 12 & 7 & 12 \end{array} \tag{11}$$

If there is only an assignment which has *0* inputs, its value is *0* and this assignment is clearly minimal. Thus, we make use of König and Hall theorem to find this type of assignments. If there is no assignment like this, algorithm provides a set where $|X|>|G(X)|$ and $X \subset O_s$. On condition that inputs are not negative, we can decrease lines of *X* in amount of some δ and increase columns of *G(X)* in amount of δ for some δ positive constants. This operation decreases value of each assignment in amount of $\delta(|X|-|G(X)|)$. This operation is maintained till δ reaches the possible size. $X = \{o_{s1}, o_{s3}, o_{s4}\}$, $G(X)=\{o_{t1}\}$ and $\delta = 5$ in the instance. In this case, new matrix is as follows:

$$\begin{array}{c|cccc} & o_{t1} & o_{t2} & o_{t3} & o_{t4} \\ \hline o_{s1} & 0 & 3 & 0 & 6 \\ o_{s2} & 5 & 0 & 0 & 0 \\ o_{s3} & 0 & 7 & 6 & 8 \\ o_{s4} & 0 & 7 & 2 & 7 \end{array} \tag{12}$$

If an assignment with *0* value cannot be founded again, it is maintained as $X = \{o_{s3}, o_{s4}\}$, $G(X) = \{o_{t1}\}$ and $\delta = 2$. New matrix is as follows:

$$\begin{array}{c|cccc} & o_{t1} & o_{t2} & o_{t3} & o_{t4} \\ \hline o_{s1} & 2 & 3 & 0 & 6 \\ o_{s2} & 7 & 0 & 0 & 0 \\ o_{s3} & 0 & 5 & 4 & 6 \\ o_{s4} & 0 & 5 & 0 & 5 \end{array}. \tag{13}$$

In the next step, $X = \{o_{s1}, o_{s3}, o_{s4}\}$, $G(X) = \{o_{t1}, o_{t3}\}$ and $\delta = 3$. The matrix is founded as follows:

$$\begin{array}{c|cccc} & o_{t1} & o_{t2} & o_{t3} & o_{t4} \\ \hline o_{s1} & 2 & 0 & 0 & 3 \\ o_{s2} & 10 & 0 & 3 & 0 \\ o_{s3} & 0 & 2 & 4 & 3 \\ o_{s4} & 0 & 2 & 0 & 2 \end{array} \tag{14}$$

As a consequent, we find an assignment as a →2, b →4, c →1, d →3. We can see here that minimum assignment in the algorithm decreases in every step until $\delta > 0$. Additionally, if all weights are integers, δ is always an integer and algorithm end in the last step. In the sample, it is assumed that $|O_s|=|O_t|=n$. But if we have an assignment problem as $|O_s|<|O_t|$, we can add void edges to O_s until it is $|O_t|-|O_s|$ weights of which is *0*. In so far, we have indicated how BGOM architecture processes step by step. At the same time, we have developed software in .Net platform which practices steps of BGOM methods. Now we will examine the results of BGOM method.

APPLICATION AND EXPERIMENTAL RESULTS

Anatomy campaign (OAEI, 2009) aims to find mapping between Anatomy of adult rat and NCI (ontology explaining human anatomy). This task is maintained between ontologies which are defined with technical terms and designed carefully in a domain which we have found highly wide. Large scale ontologies shows difference from other ontologies in terms of their limited conceptualization, heavy usage of their relation parts and usage of particular information notes and roles. OAEI includes carefully analyzed techniques based on complex and medical information together with many mapping techniques like simple text matching techniques (OAEI, 2009). Anatomy campaign consists of 4 Subparts. Subpart 1 is obligatory for all participants. Subparts 2, 3 and 4 are optional. Subparts 1, 2 and 3 are tasks of standard mapping of two ontologies. Subpart 4 is just added to the campaign. In this part, reference ontology is given as input together with two ontologies. In all these Subparts, mapping systems should produce a matching between rat and human ontologies changing according to precision and recall values. Matching produced for Subpart 1 should both be possible for precision and recall, and be an optimal solution. OAEI campaign focuses on f-measure value (OAEI, 2009).

We have participated in Subpart 1 which is compulsory in OAEI with BGOM architecture we have developed and we obtained good results. BGOM architecture applies the steps below in Subpart 1.

1. Two ontologies are modeled to apply the algorithm and we obtain a score matrix from these.
2. BGOM uses Kuhn-Munkres algorithm to find optimal map. After score matrix is calculated, it is given as input to Kuhn-Munkres algorithm to find optimal map.
3. Iteration maintains until real world objects that correspond to each other most are optimally matched.
4. After the iteration, concepts between source and target ontologies are optimally mapped.

Test data set

One of the ontology data sets that we used in BGOM, the dictionary defining human anatomy which is prepared by architecture is National Cancer Institute (NCI) and the other is the one which is developed in frame of adult rat gene anatomy database. Both sources are open biomedical ontologies (OBO). Human anatomy ontology consists of 3304 anatomic concepts (classes) and rat anatomy ontology consists of 2744 anatomic concepts (classes), these sets are fairly big sets. Classes in ontology are indicated with their owl: class labels and equivalents. BGOM modem makes use of labels and equivalents in score matrix establishment. BGOM system has given fairly good results although it has not use domain background information.

Application

We have used 4 GB RAM Intel ® Core(TM)2 Duo CPU P9600 @ 2.66 GHz to test BGOM model recommended in this thesis and developed an ontology mapping software on .Net 3.5 platform. This software practices the steps that follow:

1. Preprocessing ontology (ontology preprocessing from RDF datasets).
2. Preparing general data matrix.
3. Correspondence calculation for score matrix and forming score matrix.
4. Application of Kuhn-Munkres bipartite graph optimal assignment algorithm and finding optimal mapping.
5. Preparing matching results as RDF format.

Preprocessing ontologies and design of general data matrix

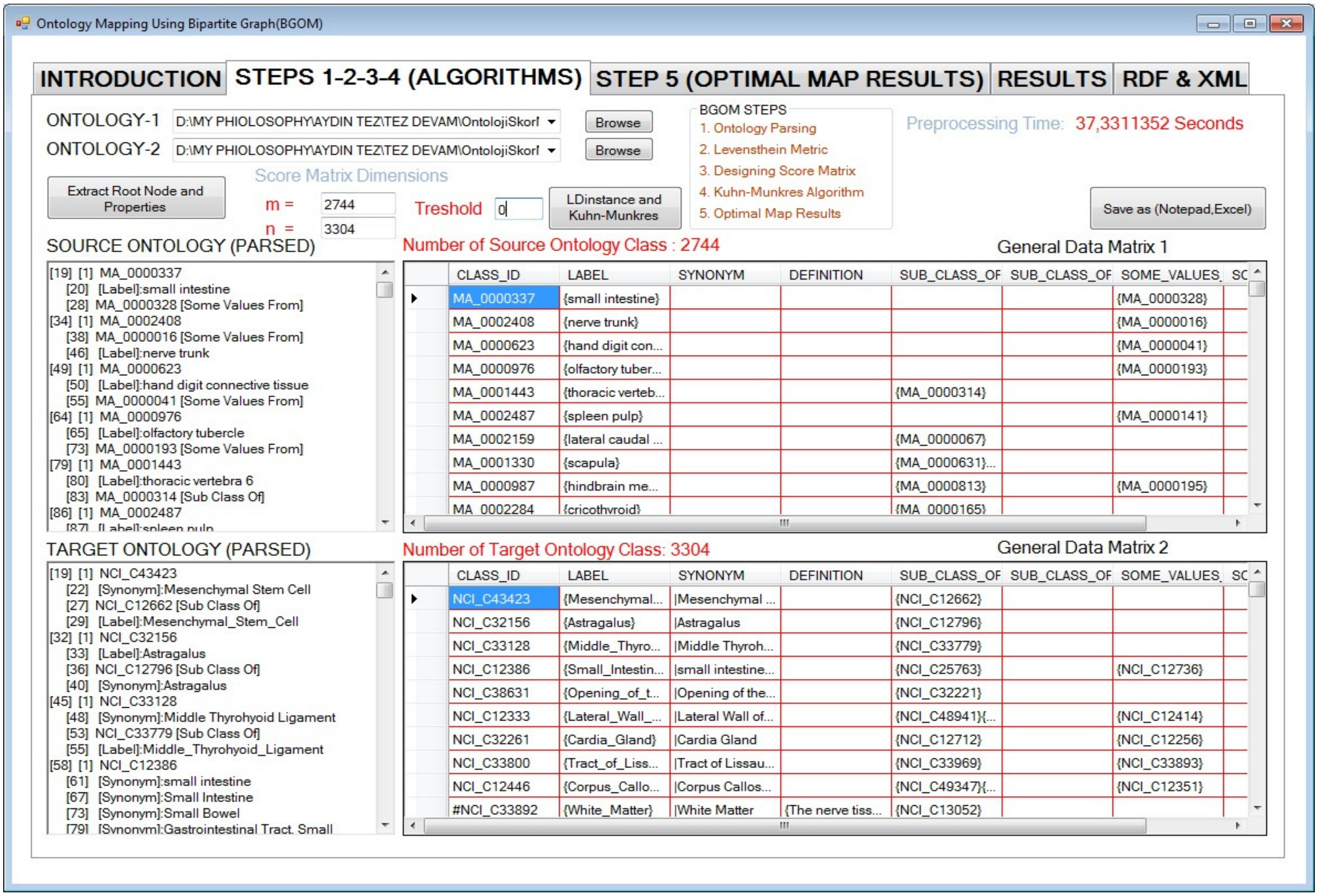
We give NCI and rat anatomy as input to BGOM system. As shown in Figure 7, BGOM system separates components like classId, labels, equivalents, sub and super classes, abbreviations and definitions included in source and target ontology datasets given in RDF format. After separation process of datasets, we obtain two general data matrixes of O_s and O_t columns of which compose of the components we mentioned earlier, are as follows:

$$G_s = \begin{pmatrix} p_s 11 & p_s 12 & \dots & p_s 1n \\ p_s 21 & p_s 22 & \dots & p_s 2n \\ \vdots & \vdots & \vdots & \vdots \\ p_s m1 & p_s m2 & \dots & p_s mn \end{pmatrix}_{m\times n} \qquad (15)$$

$$G_t = \begin{pmatrix} p_t 11 & p_t 12 & \dots & p_t 1n \\ p_t 21 & p_t 22 & \dots & p_t 2n \\ \vdots & \vdots & \vdots & \vdots \\ p_t m1 & p_t m2 & \dots & p_t mn \end{pmatrix}_{m\times n} \qquad (16)$$

Levensthein distance, score matrix and Kuhn-Munkres algorithm

We should compose score matrixes from data matrixes. We have used Levensthein metric to calculate pure synthetic correspondence between classes. We obtained our score matrix after this process. We have given score matrix that we have obtained in the last step to Kuhn-Munkres optimal assignment algorithm as input. BGOM system maintains iteration until optimal matching results are founded and produces optimal mapping input as shown.


Ontology Mapping Using Bipartite Graph(BGOM)
INTRODUCTION
STEPS 1-2-3-4 (ALGORITHMS)
STEP 5 (OPTIMAL MAP RESULTS)
RESULTS
RDF & XML
Browse
Browse
Extract Root Node and Properties
LDinstance and Kuhn-Munkres
Save as (Notepad,Excel)

ONTOLOGY-1 D:\MY PHIOLOSOPHY\AYDIN TEZ\TEZ DEVAM\OntolojiSkorl

ONTOLOGY-2 D:\MY PHIOLOSOPHY\AYDIN TEZ\TEZ DEVAM\OntolojiSkorl

Score Matrix Dimensions

m = 2744

n = 3304

Treshold 0

BGOM STEPS

1. Ontology Parsing
2. Levensthein Metric
3. Designing Score Matrix
4. Kuhn-Munkres Algorithm
5. Optimal Map Results

Preprocessing Time: 37,3311352 Seconds

SOURCE ONTOLOGY (PARSED)

```
[19] [1] MA_0000337
   [20] [Label]:small intestine
   [28] MA_0000328 [Some Values From]
[34] [1] MA_0002408
   [38] MA_0000016 [Some Values From]
   [46] [Label]:nerve trunk
[49] [1] MA_0000623
   [50] [Label]:hand digit connective tissue
   [55] MA_0000041 [Some Values From]
[64] [1] MA_0000976
   [65] [Label]:olfactory tubercle
   [73] MA_0000193 [Some Values From]
[79] [1] MA_0001443
   [80] [Label]:thoracic vertebra 6
   [83] MA_0000314 [Sub Class Of]
[86] [1] MA_0002487
   [87] [Label]:spleen pulp
```

Number of Source Ontology Class : 2744

General Data Matrix 1

CLASS_ID	LABEL	SYNONYM	DEFINITION	SUB_CLASS_OF	SUB_CLASS_OF	SOME_VALUES_	SC
MA_0000337	{small intestine}					{MA_0000328}	
MA_0002408	{nerve trunk}					{MA_0000016}	
MA_0000623	{hand digit con...					{MA_0000041}	
MA_0000976	{olfactory tuber...					{MA_0000193}	
MA_0001443	{thoracic verteb...			{MA_0000314}			
MA_0002487	{spleen pulp}					{MA_0000141}	
MA_0002159	{lateral caudal ...			{MA_0000067}			
MA_0001330	{scapula}			{MA_0000631}...			
MA_0000987	{hindbrain me...			{MA_0000813}		{MA_0000195}	
MA_0002284	{cricothyroid}			{MA_0000165}			

TARGET ONTOLOGY (PARSED)

```
[19] [1] NCI_C43423
   [22] [Synonym]:Mesenchymal Stem Cell
   [27] NCI_C12662 [Sub Class Of]
   [29] [Label]:Mesenchymal_Stem_Cell
[32] [1] NCI_C32156
   [33] [Label]:Astragalus
   [36] NCI_C12796 [Sub Class Of]
   [40] [Synonym]:Astragalus
[45] [1] NCI_C33128
   [48] [Synonym]:Middle Thyrohyoid Ligament
   [53] NCI_C33779 [Sub Class Of]
   [55] [Label]:Middle_Thyrohyoid_Ligament
[58] [1] NCI_C12386
   [61] [Synonym]:small intestine
   [67] [Synonym]:Small Intestine
   [73] [Synonym]:Small Bowel
   [79] [Synonym]:Gastrointestinal Tract, Small
```

Number of Target Ontology Class: 3304

General Data Matrix 2

CLASS_ID	LABEL	SYNONYM	DEFINITION	SUB_CLASS_OF	SUB_CLASS_OF	SOME_VALUES_	SC
NCI_C43423	{Mesenchymal...	|Mesenchymal ...		{NCI_C12662}			
NCI_C32156	{Astragalus}	|Astragalus		{NCI_C12796}			
NCI_C33128	{Middle_Thyro...	|Middle Thyroh...		{NCI_C33779}			
NCI_C12386	{Small_Intestin...	|small intestine...		{NCI_C25763}		{NCI_C12736}	
NCI_C38631	{Opening_of_t...	|Opening of the...		{NCI_C32221}			
NCI_C12333	{Lateral_Wall_...	|Lateral Wall of...		{NCI_C48941}{...		{NCI_C12414}	
NCI_C32261	{Cardia_Gland}	|Cardia Gland		{NCI_C12712}		{NCI_C12256}	
NCI_C33800	{Tract_of_Liss...	|Tract of Lissau...		{NCI_C33969}		{NCI_C33893}	
NCI_C12446	{Corpus_Callo...	|Corpus Callos...		{NCI_C49347}{...		{NCI_C12351}	
#NCI_C33892	{White_Matter}	|White Matter	{The nerve tiss...	{NCI_C13052}			

Figure 7. Preprocessing ontologies.

Evaluation criteria

We follow evaluation criteria used in 2009 OAEI ontology mapping campaign to assess the correctness of mapping results produced by BGOM system. Evaluation is based upon the results provided by participants. All results covers a set composed by matching pairs and these pairs are represented as in Figure 8 and 9. Matching criteria of two classes in source ontology and target ontology are indicated in Figure 9. Here relation criterion is taken as "=", correspondence value constant as "float" and correspondence value as "1.0". There are two types of evaluation in OAEI campaign. First is benchmark test which are open tests, the results of which are known by participants. Second is blank test the results of which are not known by participants. For all tests, standard information retrieval evaluation criterions and precision, recall and f-measure values are calculated opposing to a reference map. For measurement combination problem, weighted harmonic average is also calculated (Ashpole et al., 2005). Precision, recall and f-measure values are defined with the equations that follow:

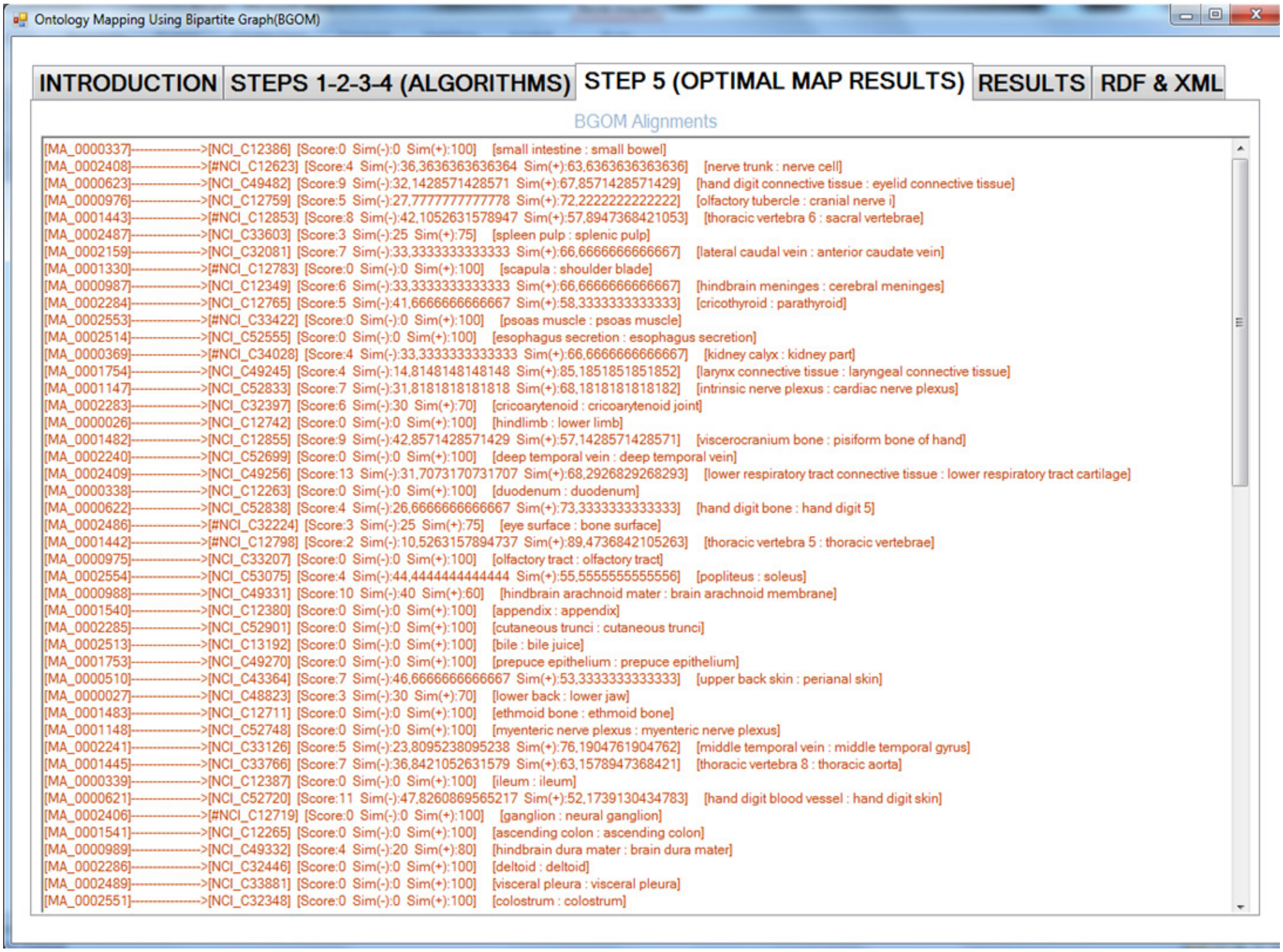
$$precision = p = \frac{correct\_found\_mappings}{all\_found\_mappings} \tag{17}$$

$$f - measure = f = \frac{2pr}{p + r} \tag{18}$$

$$recal = r = \frac{correct\_found\_mappings}{all\_possible\_mappings} \tag{19}$$

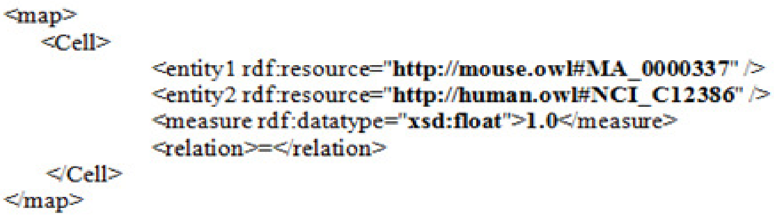
Runtime and performance

We have mentioned that there are 3304 concepts in human anatomy ontology and 2744 concepts in rat anatomy ontology. That is, system aims to find the most suitable 2744 matches among approximately 9 million matches. A correctness threshold value will be used to avoid unnecessary and wrong matches. This process


Ontology Mapping Using Bipartite Graph(BGOM)
INTRODUCTION
STEPS 1-2-3-4 (ALGORITHMS)
STEP 5 (OPTIMAL MAP RESULTS)
RESULTS
RDF & XML
BGOM Alignments

```
[MA_0000337]----------------->[NCI_C12386]  [Score:0  Sim(-):0  Sim(+):100]    [small intestine : small bowel]
[MA_0002408]----------------->[#NCI_C12623]  [Score:4  Sim(-):36,3636363636364  Sim(+):63,6363636363636]    [nerve trunk : nerve cell]
[MA_0000623]----------------->[NCI_C49482]  [Score:9  Sim(-):32,1428571428571  Sim(+):67,8571428571429]    [hand digit connective tissue : eyelid connective tissue]
[MA_0000976]----------------->[NCI_C12759]  [Score:5  Sim(-):27,7777777777778  Sim(+):72,2222222222222]    [olfactory tubercle : cranial nerve i]
[MA_0001443]----------------->[#NCI_C12853]  [Score:8  Sim(-):42,1052631578947  Sim(+):57,8947368421053]    [thoracic vertebra 6 : sacral vertebrae]
[MA_0002487]----------------->[NCI_C33603]  [Score:3  Sim(-):25  Sim(+):75]    [spleen pulp : splenic pulp]
[MA_0002159]----------------->[NCI_C32081]  [Score:7  Sim(-):33,3333333333333  Sim(+):66,6666666666667]    [lateral caudal vein : anterior caudate vein]
[MA_0001330]----------------->[#NCI_C12783]  [Score:0  Sim(-):0  Sim(+):100]    [scapula : shoulder blade]
[MA_0000987]----------------->[NCI_C12349]  [Score:6  Sim(-):33,3333333333333  Sim(+):66,6666666666667]    [hindbrain meninges : cerebral meninges]
[MA_0002284]----------------->[NCI_C12765]  [Score:5  Sim(-):41,6666666666667  Sim(+):58,3333333333333]    [cricothyroid : parathyroid]
[MA_0002553]----------------->[#NCI_C33422]  [Score:0  Sim(-):0  Sim(+):100]    [psoas muscle : psoas muscle]
[MA_0002514]----------------->[NCI_C52555]  [Score:0  Sim(-):0  Sim(+):100]    [esophagus secretion : esophagus secretion]
[MA_0000369]----------------->[#NCI_C34028]  [Score:4  Sim(-):33,3333333333333  Sim(+):66,6666666666667]    [kidney calyx : kidney part]
[MA_0001754]----------------->[NCI_C49245]  [Score:4  Sim(-):14,8148148148148  Sim(+):85,1851851851852]    [larynx connective tissue : laryngeal connective tissue]
[MA_0001147]----------------->[NCI_C52833]  [Score:7  Sim(-):31,8181818181818  Sim(+):68,1818181818182]    [intrinsic nerve plexus : cardiac nerve plexus]
[MA_0002283]----------------->[NCI_C32397]  [Score:6  Sim(-):30  Sim(+):70]    [cricoarytenoid : cricoarytenoid joint]
[MA_0000026]----------------->[NCI_C12742]  [Score:0  Sim(-):0  Sim(+):100]    [hindlimb : lower limb]
[MA_0001482]----------------->[NCI_C12855]  [Score:9  Sim(-):42,8571428571429  Sim(+):57,1428571428571]    [viscerocranium bone : pisiform bone of hand]
[MA_0002240]----------------->[NCI_C52699]  [Score:0  Sim(-):0  Sim(+):100]    [deep temporal vein : deep temporal vein]
[MA_0002409]----------------->[NCI_C49256]  [Score:13  Sim(-):31,7073170731707  Sim(+):68,2926829268293]    [lower respiratory tract connective tissue : lower respiratory tract cartilage]
[MA_0000338]----------------->[NCI_C12263]  [Score:0  Sim(-):0  Sim(+):100]    [duodenum : duodenum]
[MA_0000622]----------------->[NCI_C52838]  [Score:4  Sim(-):26,6666666666667  Sim(+):73,3333333333333]    [hand digit bone : hand digit 5]
[MA_0002486]----------------->[#NCI_C32224]  [Score:3  Sim(-):25  Sim(+):75]    [eye surface : bone surface]
[MA_0001442]----------------->[#NCI_C12798]  [Score:2  Sim(-):10,5263157894737  Sim(+):89,4736842105263]    [thoracic vertebra 5 : thoracic vertebrae]
[MA_0000975]----------------->[NCI_C33207]  [Score:0  Sim(-):0  Sim(+):100]    [olfactory tract : olfactory tract]
[MA_0002554]----------------->[NCI_C53075]  [Score:4  Sim(-):44,4444444444444  Sim(+):55,5555555555556]    [popliteus : soleus]
[MA_0000988]----------------->[NCI_C49331]  [Score:10  Sim(-):40  Sim(+):60]    [hindbrain arachnoid mater : brain arachnoid membrane]
[MA_0001540]----------------->[NCI_C12380]  [Score:0  Sim(-):0  Sim(+):100]    [appendix : appendix]
[MA_0002285]----------------->[NCI_C52901]  [Score:0  Sim(-):0  Sim(+):100]    [cutaneous trunci : cutaneous trunci]
[MA_0002513]----------------->[NCI_C13192]  [Score:0  Sim(-):0  Sim(+):100]    [bile : bile juice]
[MA_0001753]----------------->[NCI_C49270]  [Score:0  Sim(-):0  Sim(+):100]    [prepuce epithelium : prepuce epithelium]
[MA_0000510]----------------->[NCI_C43364]  [Score:7  Sim(-):46,6666666666667  Sim(+):53,3333333333333]    [upper back skin : perianal skin]
[MA_0000027]----------------->[NCI_C48823]  [Score:3  Sim(-):30  Sim(+):70]    [lower back : lower jaw]
[MA_0001483]----------------->[NCI_C12711]  [Score:0  Sim(-):0  Sim(+):100]    [ethmoid bone : ethmoid bone]
[MA_0001148]----------------->[NCI_C52748]  [Score:0  Sim(-):0  Sim(+):100]    [myenteric nerve plexus : myenteric nerve plexus]
[MA_0002241]----------------->[NCI_C33126]  [Score:5  Sim(-):23,8095238095238  Sim(+):76,1904761904762]    [middle temporal vein : middle temporal gyrus]
[MA_0001445]----------------->[NCI_C33766]  [Score:7  Sim(-):36,8421052631579  Sim(+):63,1578947368421]    [thoracic vertebra 8 : thoracic aorta]
[MA_0000339]----------------->[NCI_C12387]  [Score:0  Sim(-):0  Sim(+):100]    [ileum : ileum]
[MA_0000621]----------------->[NCI_C52720]  [Score:11  Sim(-):47,8260869565217  Sim(+):52,1739130434783]    [hand digit blood vessel : hand digit skin]
[MA_0002406]----------------->[#NCI_C12719]  [Score:0  Sim(-):0  Sim(+):100]    [ganglion : neural ganglion]
[MA_0001541]----------------->[NCI_C12265]  [Score:0  Sim(-):0  Sim(+):100]    [ascending colon : ascending colon]
[MA_0000989]----------------->[NCI_C49332]  [Score:4  Sim(-):20  Sim(+):80]    [hindbrain dura mater : brain dura mater]
[MA_0002286]----------------->[NCI_C32446]  [Score:0  Sim(-):0  Sim(+):100]    [deltoid : deltoid]
[MA_0002489]----------------->[NCI_C33881]  [Score:0  Sim(-):0  Sim(+):100]    [visceral pleura : visceral pleura]
[MA_0002551]----------------->[NCI_C32348]  [Score:0  Sim(-):0  Sim(+):100]    [colostrum : colostrum]
```

Figure 8. Results of BGOM optimal mapping.



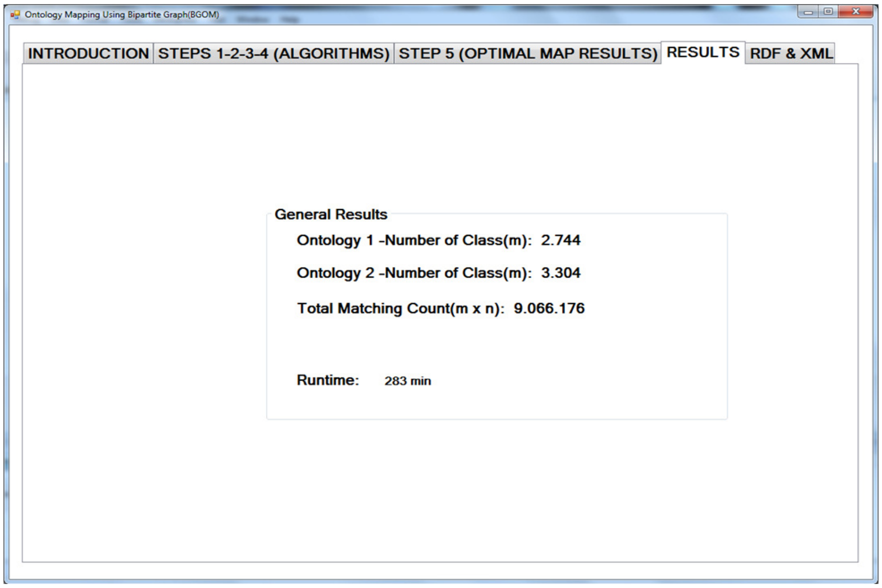
```
<map>
    <Cell>
              <entity1 rdf:resource="http://mouse.owl#MA_0000337" />
              <entity2 rdf:resource="http://human.owl#NCI_C12386" />
              <measure rdf:datatype="xsd:float">1.0</measure>
              <relation>=</relation>
    </Cell>
</map>
```

Figure 9. Representation of correspondence of two classes.

takes approximately 4 h, 43 min and 46 s. System has a $O(n^2)$ working time because optimal matching algorithm works on 2-dimension matrix. The results of the developed architecture are shown in Figure 10. Figure 11 shows the output format requested by OAEI.

Comparison of BGOM method with other systems

We have sent electronically the map we obtained to OAEI in RDF format to measure the success of BGOM system. Matching results are calculated by OAEI and informed to


Ontology Mapping Using Bipartite Graph(BGOM)
INTRODUCTION
STEPS 1-2-3-4 (ALGORITHMS)
STEP 5 (OPTIMAL MAP RESULTS)
RESULTS
RDF & XML

General Results

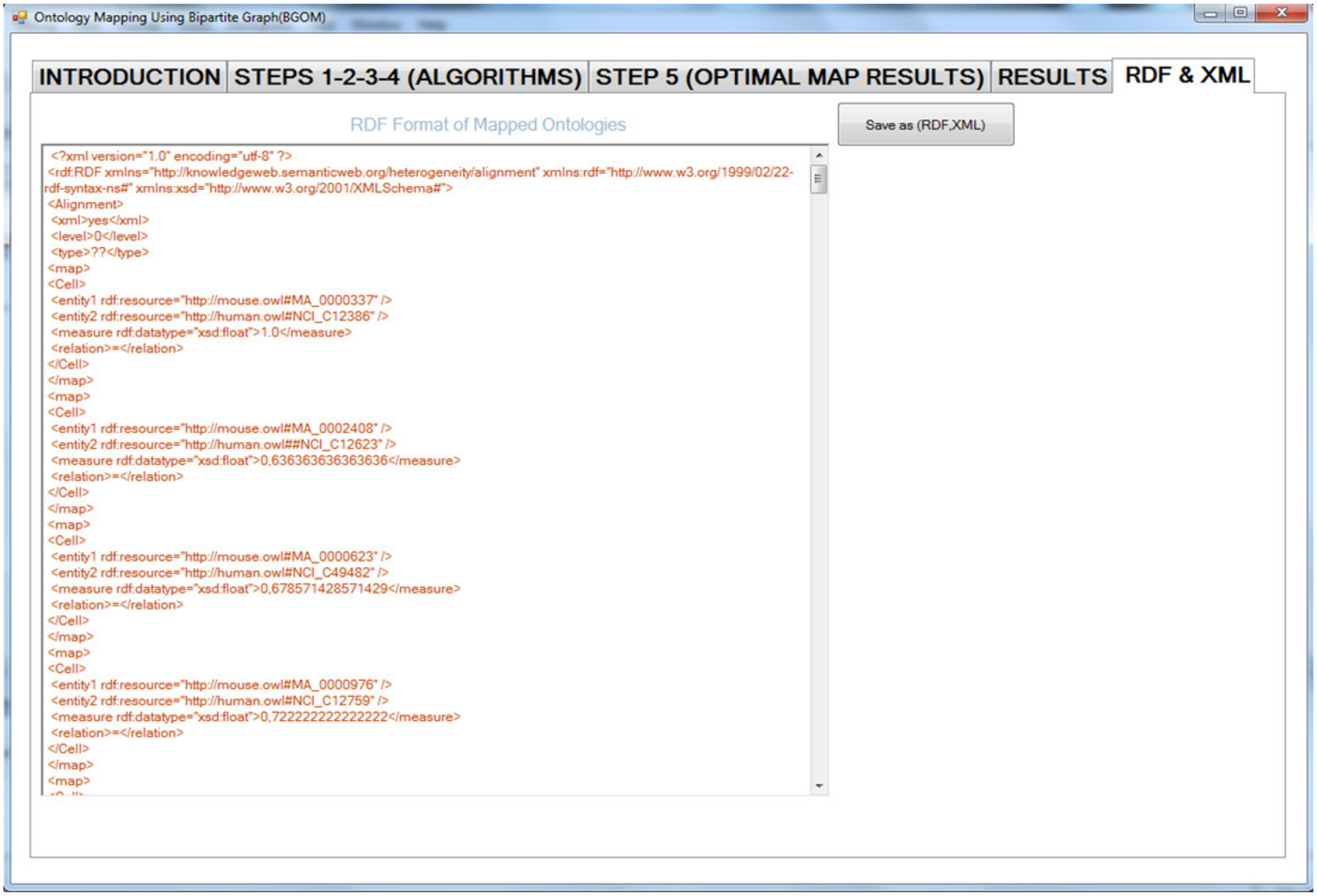
Ontology 1 -Number of Class(m): 2.744

Ontology 2 -Number of Class(m): 3.304

Total Matching Count(m x n): 9.066.176

Runtime: 283 min

Figure 10. Mapping results.


Ontology Mapping Using Bipartite Graph(BGOM)
INTRODUCTION
STEPS 1-2-3-4 (ALGORITHMS)
STEP 5 (OPTIMAL MAP RESULTS)
RESULTS
RDF & XML
Save as (RDF,XML)

RDF Format of Mapped Ontologies

```
 <?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF xmlns="http://knowledgeweb.semanticweb.org/heterogeneity/alignment" xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#" xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
<Alignment>
 <xml>yes</xml>
 <level>0</level>
 <type>??</type>
<map>
<Cell>
 <entity1 rdf:resource="http://mouse.owl#MA_0000337" />
 <entity2 rdf:resource="http://human.owl#NCI_C12386" />
 <measure rdf:datatype="xsd:float">1.0</measure>
 <relation>=</relation>
</Cell>
</map>
<map>
<Cell>
 <entity1 rdf:resource="http://mouse.owl#MA_0002408" />
 <entity2 rdf:resource="http://human.owl##NCI_C12623" />
 <measure rdf:datatype="xsd:float">0,636363636363636</measure>
 <relation>=</relation>
</Cell>
</map>
<map>
<Cell>
 <entity1 rdf:resource="http://mouse.owl#MA_0000623" />
 <entity2 rdf:resource="http://human.owl#NCI_C49482" />
 <measure rdf:datatype="xsd:float">0,678571428571429</measure>
 <relation>=</relation>
</Cell>
</map>
<map>
<Cell>
 <entity1 rdf:resource="http://mouse.owl#MA_0000976" />
 <entity2 rdf:resource="http://human.owl#NCI_C12759" />
 <measure rdf:datatype="xsd:float">0,722222222222222</measure>
 <relation>=</relation>
</Cell>
</map>
<map>
```

Figure 11. BGOM mapping result in RDF format.

Table 1. Comparison of OAEI 2007 anatomy results to BGOM.

System	Runtime	Precision	Recall	F-Measure	Recall+
AOAS	n.a	0.928	0.804	0.861	0.505
SAMBO	360 min	0.845	0.786	0.815	0.580
BGOM	283 min	0.840	0.740	0.790	0.360
Label Eq.	180 min	0.987	0.605	0.750	0.000
ASMOV	900 min	0.803	0.701	0.749	0.270
Falcon-AO	12 min	0.964	0.591	0.733	0.123
TaxoMap	300 min	0.596	0.732	0.657	0.230
Agreementmaker	30 min	0.558	0.635	0.594	0.262
Prior+	23 min	0.594	0.590	0.592	0.338
Lily	4 days	0.481	0.559	0.517	0.374
RiMOM	240 min	0.377	0.659	0.480	0.390
X-SOM	600 min	0.916	0.248	0.390	0.008
DSSim	75 min	0.208	0.187	0.197	0.067

us as precision = 0.840, recall = 0.740, f-measure = 0.790 and recall+ = 0.360. These results have shown us that success of BGOM system is not that much far away from the best four systems. Comparison of BGOM architecture to other systems according to the series reducing according to f-measure is indicated in the tables that follow. Eleven systems in total participated in anatomy part of OAEI in 2007. This system can be roughly divided into three groups. Type A systems are professional in biomedical ontology mapping and make use of biomedical "domain background" information mostly. These systems are AOAS and Sambo. Type B systems can solve the problem in matching of different domains but also include a component of biomedical information (UMLS usage as syntactic reference system). ASMOV and RiMOM are included in this category. Type C systems are considered as versatile matching systems which cannot make a differentiation between medical ontologies or ontologies of different domains. In this respect, BGOM system is clearly a type C system. Although BGOM does not use "domain background" information, it is clearly seen in Table 1 that it has fairly good results. Working time for SAMBO and ASMOV are confused due to some failures in evaluation process. SAMBO have obtained results approximately in 360 min for Part 1 and ASMOV in 900 min (it was 7 days according to the previous test results). BGOM system has completed the whole process in 238 min. Figure 12 shows the success of systems in order of decreasing f-measure. Working time of systems shows an essential difference. (Runtime periods are provided by participants. All matches are performed by standard PCs with similar hardware. Advantages based on hardware differences can be missed due to working time differences). Type C systems are more successful than the systems using medical information. Falcon-OA is a system which solves big matching problem by applying a partite based block matching strategy. Compared to other systems of type C, it completes matching task without loss of quality in terms of result matching in approximately 12 min. If similar methods can be applied to the systems like ASMOV and Lily to solve the problems about working time, these systems can be considered, too. BGOM provides a good working time compared to other participants.

Earlier mentioned systems in total participated in anatomy task. These systems divide into two groups: a group which uses biomedical information (UMLS as syntactic reference system) and another group which does not use a particular information domain. Systems in the first group are SAMBO and ASMOV and the other systems are in the second group. Table 2 indicates a general view on participant systems. In 2007, first group systems, especially the best three systems using information (AOAS, SAMBO and ASMOV) have an important advantage of finding unusual matches. Table 2 makes a list of results of participant in e-sequence reducing in terms of f-measure obtained (OAEI have regulated the systems according to f-measure values in 2008 and used the values of 2007 of this systems which did not participate this year). SAMBO and SAMBOdtf which is a version of it in the first place of 2008 campaign. Compared to the campaign in 2007, SAMBO has obtained better results for precision and recall values in 2008. SAMBO has nearly reached to f-measure value which AOAS obtained in 2007. This is not a considerable result. Because SAMBO is a system based upon choosing matching strategies with human interaction, while SAMBO and SAMBOdtf focus on biomedical information usage. RiMOM matching system is based on calculating label edit-distance value in accordance with correspondence diffusion strategies. Due to an important improvement in results of RiMOM, it is one of the best

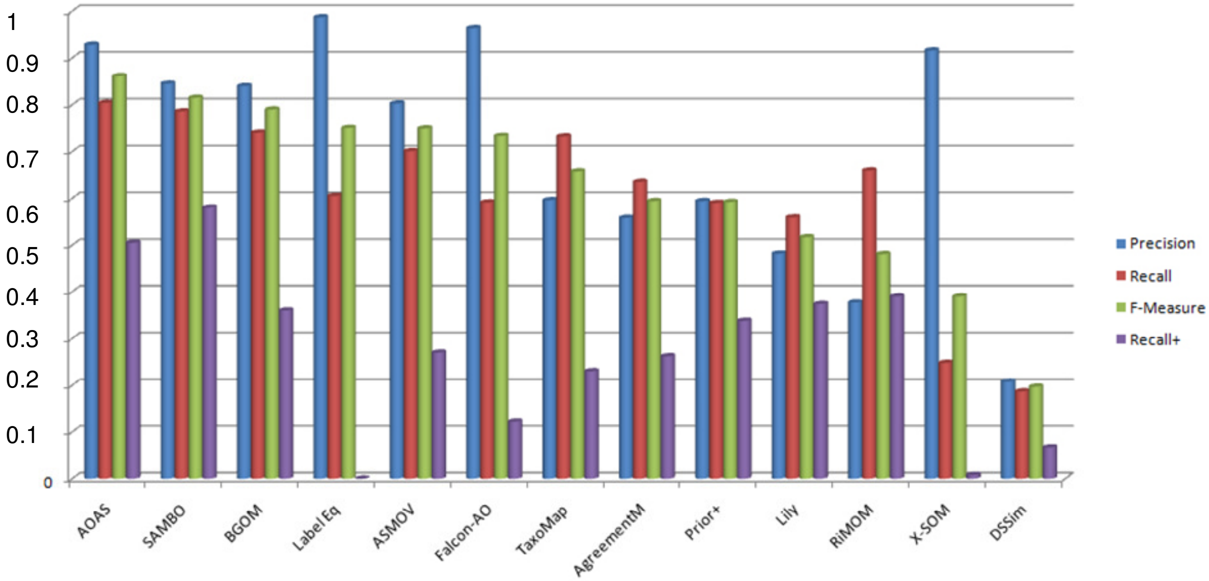


Figure 12. Comparison of OAEI 2007 anatomy results to BGOM.

Table 2. Comparison of OAEI 2008 anatomy results to BGOM.

System	Runtime	Precision	Recall	F-Measure	Recall+
AOAS	n.a	0.928	0.815	0.868	0.523
SAMBO	720 min	0.869	0.836	0.852	0.586
SAMBOdtf	1020 min	0.831	0.833	0.832	0.579
RiMOM	24 min	0.929	0.735	0.821	0.350
BGOM	283 min	0.840	0.740	0.790	0.360
Aflood	1 min 5 s	0.874	0.682	0.766	0.275
Label Eq.	-	0.981	0.613	0.755	0.000
Lily	200 min	0.796	0.693	0.741	0.387
FalconAO	12 min	0.963	0.599	0.738	0.127
ASMOV	230 min	0.787	0.652	0.713	0.246
AROMA	3 min 5 s	0.803	0.560	0.660	0.302
TaxoMap	25 min	0.460	0.764	0.638	0.234
DSSim	17 min	0.616	0.624	0.620	0.170
Prior	23 min	0.593	0.598	0.596	0.350
XSOM	600 min	0.915	0.212	0.344	0.008

matching systems for anatomy task although it does not use a particular information. In addition, it should not be forgotten that RiMOM has completed matching tasks in a fairly effective way. Nearly all matching systems in 2007 have accomplished their own results but ASMOV and TaxoMap have had worse results when compared. Furthermore, BGOM is an edit-distance based mapping system and can complete matching tasks in an effective way.

Figure 13 shows the success of systems in order of decreasing f-measure. Although, the matches accepted have been done in different machines, OAEI thinks that working time submitted is not a very distinguishing criteria and these values given by participants can be used in matching. Only for Aflood and AROMA which are the fastest two systems, working time has been measured in same machine (Pentium D 3.4 GHz, 2 GB RAM). Compared to the working times measured in previous years, the fastest working time in 2007 was of Lily and ASMOV. It can be seen that these systems have

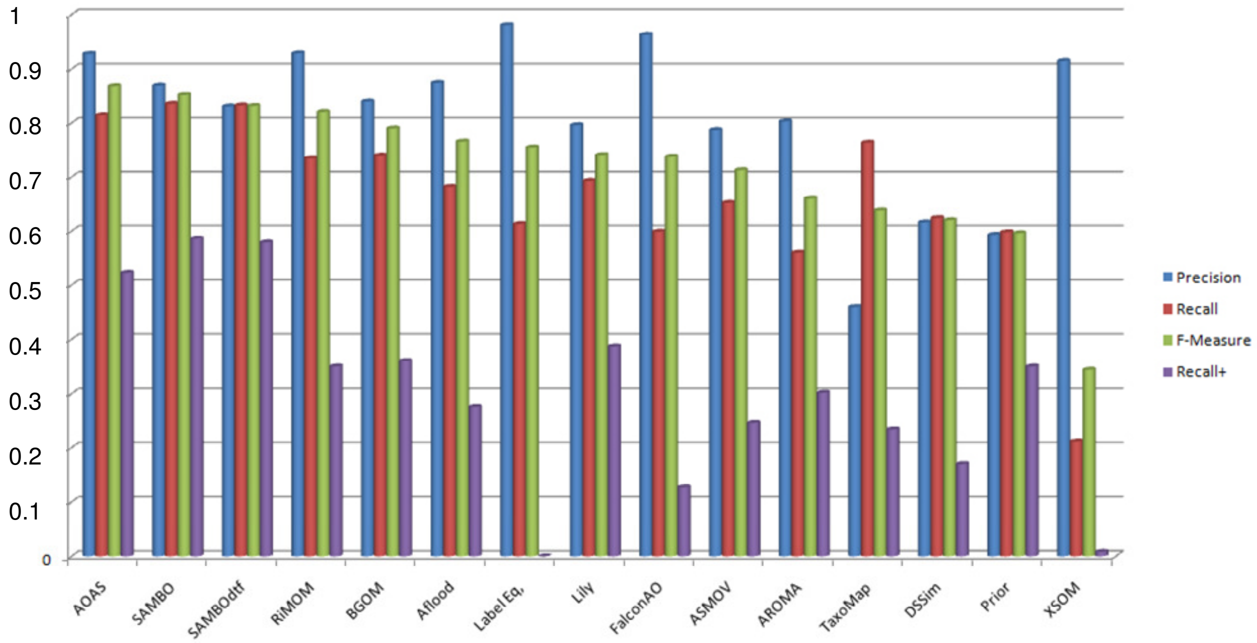


Figure 13. Comparison of OAEI 2008 anatomy results to BGOM.

Table 3. Comparison of OAEI 2009 anatomy results to BGOM.

System	Runtime (min)	Precision	Recall	F-Measure	Recall+
SOBOM	19	0.952	0.777	0.855	0.431
RiMOM	10	0.940	0.684	0.792	0.183
BGOM	283	0.840	0.740	0.790	0.360
TaxoMap	12	0.870	0.678	0.762	0.222
DSSim	12	0.853	0.676	0.754	0.185
ASMOV	4	0.746	0.755	0.751	0.419
Aflood	15 s/4	0.873	0.653	0.747	0.197
Lily	99	0.738	0.739	0.739	0.477
Agreementmaker	23	0.865	0.798	0.731	0.489
Aroma	1	0.775	0.678	0.723	0.368
Kosimap	5	0.866	0.619	0.722	0.154

considerably decreased working time this year. Among all systems, AROMA and Aflood, both of which have participated for the first time, have shown the best performance in the most effective way in terms of operation times. As shown in Figure 13 especially Aflood and BGOM systems obtain effectively high-quality results.

Table 3 makes a list of participant in a reducing sequence in terms of f-measure obtained for Subpart 1 in anatomy campaign. In the first two places, we find SOBOM and agreementmaker. Both systems have obtained fairly good results. SOBOM has obtained the best result although it participated for the first time in 2009 (OAEI, 2009). AgreementMaker forms few certain matches but finds more correct matches. No system could establish important matching pairs for Subpart 1 in 2009 (OAEI, 2009). When compared with the year 2008, RiMOM system is worse comparatively in terms of f-measure. Precision value has increased but this causes loss of recall, especially an important loss of racall+. Systems listed in Table 3, obtains similar results in terms of matching quality (f-measure changing between 0.72 and 0.76). But there could be important differences between precision and recall. All systems except for ASMOV

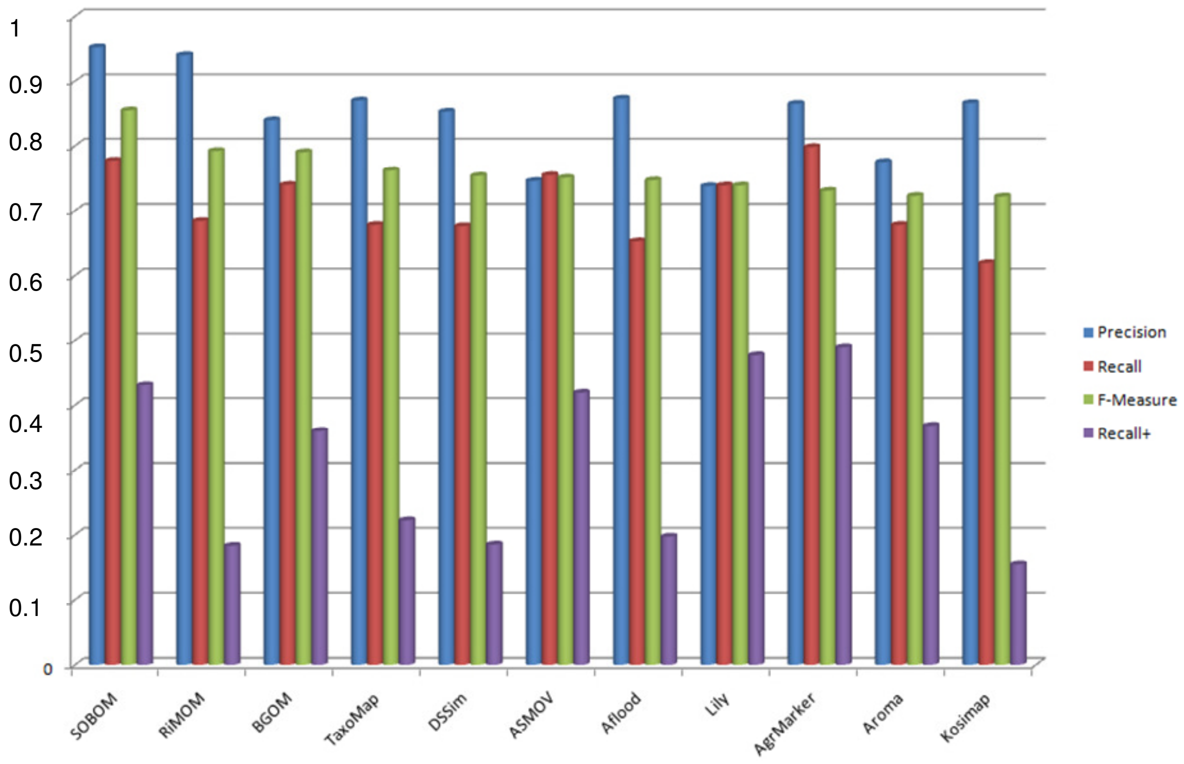


Figure 14. Comparison of OAEI 2009 anatomy results to BGOM.

and Lily approve certainty on recall. It should be known that 0.755 f-measure value can be easily obtained by forming a complete matching on condition of finding non-simple matching pairs. Additionally, finding 0.755 f-measure value for a recall value required on precision is comparatively difficult. Therefore, results of ASMOV and Lily should be interpreted positively in terms of f-measure (OAEI, 2009). Evaluation results for Aflood require additional explanations. Aflood works with a configuration resulted in an important reduction in working time (15 s). Figure 14 shows the success of systems in order of decreasing f-measure. Due to OAEI evaluation process, the accepted matches are formed by participants that accomplished their systems in their own machines. However, resulted working hour measurements submit an approximate ground for a reasonable matching. OAEI observed important differences in terms of specified working times in 2007 (OAEI, 2009). While Lily required several days to complete matching task, more than half of the systems could not match ontologies in a period less than an hour. OAEI has already observed remaining working time and the evaluation of this year showed that only one system required more than an hour. The fastest system in the last campaign is Aflood (15 s). Then comes Aroma and it completes matching approximately in 1 min. Table 1, 2 and 3 indicates that BGOM has very good results and is not that far away from the systems of the best 4 participants of evaluation campaign in 2007, 2008 and 2009.

Conclusion

BGOM system that we have developed have obtained fairly good recall, precision and f-measure results compared to other systems although it is the first time it has been evaluated by OAEI campaign. In Table 1 and 2, working time of BGOM is good but working time of other systems are better in 2009 because they have improved their algorithms in three years. We have presented architecture of ontologies modeled as weighted bipartite graph and we have aimed to find one-to-one optimal mapping between ontology classes which are on the same or similar domains. We have used Kuhn-Munkres optimal assignment algorithm, which is one of bipartite graph matching algorithms, to find optimal mapping between ontologies. Results of BGOM architecture have been evaluated by OAEI campaign. Although we have not used biomedical domain background information and we have been a new participant in this evaluation, we have shown that our system is fairly good with the mapping we obtained.

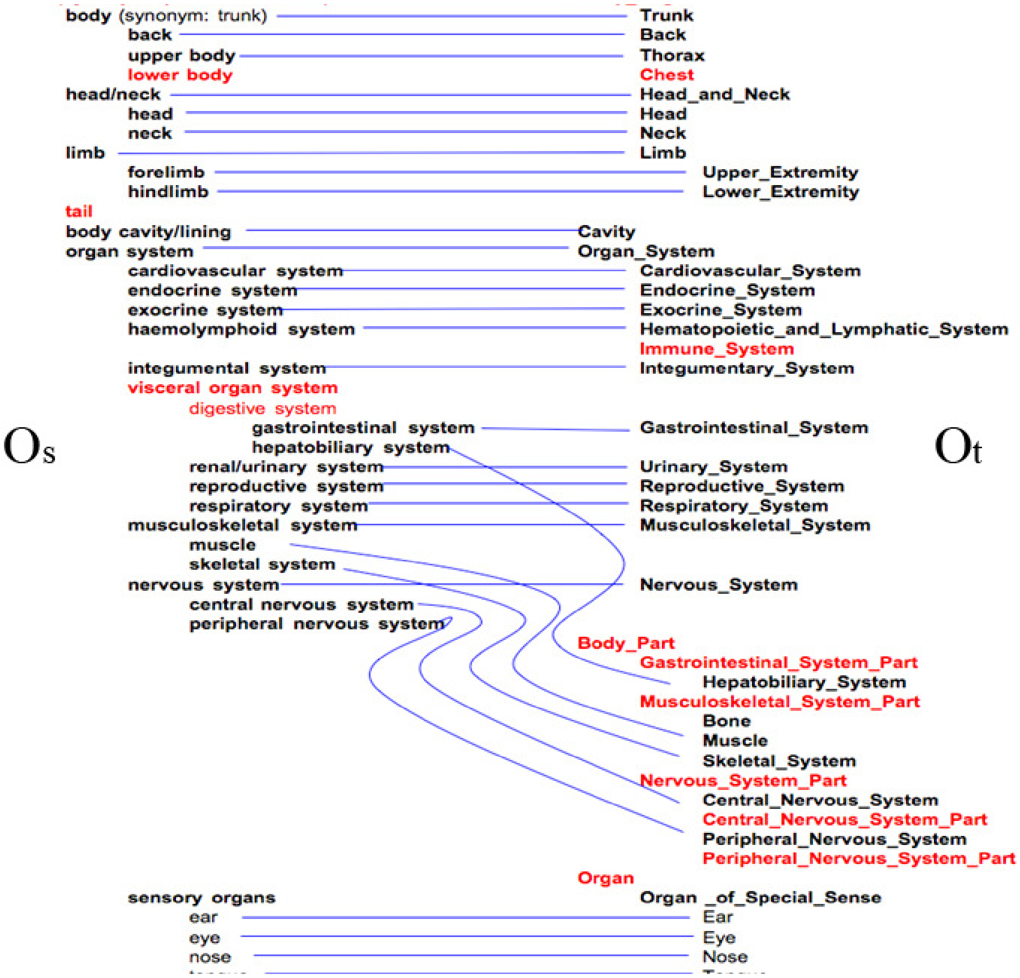


Figure 15. BGOM mapping results.

Quality and correctness of mapping formed by BGOM is directly connected to score matrix. This matrix is very important for BGOM system. There are many methods to calculate score matrix and we have used Levensthein metric to find out same. The better the strategies and methods chosen for forming score matrix, the better the results of mapping. This paper indicates the truth that ontologies in same or similar domains can be modeled as bipartite graph and a mapping system can be developed with the help of graph matching algorithms. Additionally, the results obtained are proved to be fairly good in conclusion of analyses when compared to other systems. Figure 15 shows the final result produced by BGOM system. As shown in figure, optimally matched to the nearest concepts.

Future work

A pure synthetic method has been developed in this study. Therefore, formal concepts of ontology has been considered to be too many and removed.Additionally, L-Distance is a totally optional approach and this is only used for finding the beginning correspondence score matrix. A different method can be used in place of this. The article especially moves on a label-based correspondence but it is considered that some structures like hierarchic structure of classes, real equivalent dictionary will be added in future work. It is also considered more certain results can be obtained this way.

ACKNOWLEDGEMENT

The authors are grateful to Christian Meilicke who evaluated the mapping results produced by BGOM architecture.

REFERENCES

Ashpole B, Ehrig M, Euzenat J, Stuckenschmidt H (2005). Relaxed

Precision and Recall for Ontology Matching, Proc. K-Cap 2005 workshop on Integrat. Ontol. Banff. (CA). pp. 25-32.

Berlin J, Motro A (2002). Database schema matching using machine learning with feature selection. In Proc. of the Conf. on Advanced Inform. Syst. Eng (CAiSE). pp. 8-13.

Berners-Lee T, Hendler J, Lassila O (2001). The semantic Web, Scientific American. 284(5): 34–43.

Bouquet P, Euzenat J, Franconi L, Serafini E, Stamou G, Tessaris S (2004). Specification of a Common Framework for Characterizing Alignment. Knowledge Web project EUIST2004507482 realizing the semantic web. pp. 6-10.

Curino C, Orsi G, Tanca L (2007). X-SOM Results for OAEI. Int. Semantic Web Conf. 2007: Busan, Korea. pp. 2-5.

David J (2009). AROMA results for OAEI. Int.Semantic Web Conf. 2009. Chantilly, VA, USA. pp. 1-2.

Do H, Rahm E (2002). COMA - A System for Flexible Combination of Schema Matching Approaches. Proceedings of Very Large Data Bases Conference (VLDB). pp.1-2, 3.

Doan A, Madhavan J, Dhamankar R, Domingos P, Halevy AY (2003). Learning to match ontologies on the semantic Web, VLDB J., 12(4): 303–319.

Hamdi F, Safar B, Niraula NB, Reynaud C (2009). TaxoMap in the OAEI alignment contest. Int. Semantic Web Conf. 2009: Chantilly. VA, USA. pp. 2-6.

Jean-Mary YR, Shironoshita EP, Kabuka MR (2009). ASMOV Results for OAEI. Int. Semantic Web Conf. 2009: Chantilly, VA, USA. pp. 1-2.

Lambrix P, Tan H, Liu Q (2008). SAMBO and SAMBOdtf Results for the Ontology Alignment Evaluation Initiative. Int. Semantic Web Conf. 2008: Karlsruhe, Germany – Ontol. Matching, pp. 2-5.

Mao M, Peng Y (2007). The PRIOR+: Results for OAEI Campaign. Int. Semantic Web Conf. 2007: Busan, Korea. pp. 1-4.

Meilicke C, Stuckenschmidt H (2007). Analyzing Mapping Extraction Approaches. Int. Semantic Web Conf.2007. Busan, Korea. pp. 1-2.

Murata T (2003). Graph Mining Approaches for the Discovery of Web Communities. Japan Sci. and Technol. Corporat. Tokyo. 151-0053.

Nagy M, Vargas-Vera M, Stolarski P (2008). DSSim Results for OAEI. International Semantic Web Conference 2008: Karlsruhe, Germany - Ontology Matching. pp. 3-8.

Noy N, Doan A (2005). Semantic Integration. AI Magazine Special Issue on Semantic Integration. 26(1):7-9.

Noy N, Musen M (2001). Anchor-PROMPT: Using non-local context for semantic matching. Proc. IJCAI 2001 workshop on ontol. and inform. sharing, Seattle (WA US). pp. 1-2.

OAEI (2009). Ontology Alignment Evaluation Initiative. Campaign. http://webrum.uni-mannheim.de/math/lski/anatomy09/.

Pirró G, Talia D (2010). UFOme: An ontology mapping system with strategy prediction capabilities, Data and Knowledge Engineering. pp. 1-10.

Rahm E, Bernstein PA (2001). A survey of approaches to automatic schema matching, VLDB J., 10(4): 334–350.

Rodríguez A, Egenhofer M (2003). Determining Semantic Similarity among Entity Classes from Different Ontologies. IEEE Trans. On Knowledge and Data Eng., 15(2): 442-456.

Sabel F, Antonelli FP, Stroe C, Keles UC, Maduko A (2009). Using AgreementMaker to Align Ontologies for OAEI. Int. Semantic Web Conference 2009: Chantilly, VA, USA. pp. 1-2.

Seddiqui MH, Aono M (2009). Anchor-Flood: Results for OAEI. Int. Semantic Web Conference 2009: Chantilly, VA, USA. pp. 2-3.

Tassa T (2007). A Hierarchical Clustering Algorithm Based on the Hungarian Method. Jacob Goldberger School of Engineering Bar-Ilan University, Israel Division of Computer Science. The Open University, Israel. pp. 3-4.

Wang P, Xu B (2009). Lily: Ontology Alignment Results for OAEI. Int. Semantic Web Conf. 2009. Chantilly, VA, USA. pp. 1-4.

Wikipedia (2011). Semantic Integration. http://en.wikipedia.org/wiki/Semantic _integration.

Xu P, Tao H, Zang T, Wang Y (2009). Alignment Results of SOBOM for OAEI. Int. Semantic Web Conf. 2009: Chantilly. VA, USA. pp. 2-4.

Zhang X, Zhong Q, Shi F, Li J, Tang Jie (2009). RiMOM Results for OAEI 2009, International Semantic Web Conference 2009: Chantilly, VA, USA - Ontology Matching. pp. 1-5.