2018 4th International Conference on Web Research (ICWR)

Ontology Alignment Using Inductive Logic Programming

[1] Hamed Karimi, [2] Ali Kamandi
School of Engineering Science, College of Engineering
University of Tehran
Tehran, Iran

[1, 2] {ha.karimi, kamandi}@ut.ac.ir

Abstract—**Ontologies are one of the important and effective parts of semantic web which constitute the infrastructure and background knowledge of this realm of web science. Finding valid mappings as much as possible between the concepts or entities of ontologies, especially for the large ones, is a prominent task to align those concepts together and finally merge and integrate their ontologies to make a general and global ontology that is smaller and more flexible in many applications of semantic web. This paper describes a new learning-based ontology mapping method in which inductive logic programming (ILP) is used to learn ontology mapping using information gathered from instances of each entity in order to make some correct and valid alignments between concepts of different ontologies.**

Keywords—ontology alignment; ontology matching; ontology mapping learning; inductive logic programming; semantic web;

I. INTRODUCTION

Ontologies play an important role on the Semantic Web. They make possible the widespread publication of machine understandable data and opening many opportunities for automated information processing. Ontologies are increasingly appeared as a key factor for enabling interoperability in heterogeneous systems and semantic web applications. Ontology mapping or matching is the task of finding semantic relationships between entities (i.e. concept, attribute and relation) of two different ontologies in order to make some valid and correct alignments between their concepts. It is required for combining distributed and heterogeneous ontologies. Developing such ontology mapping has been a core issue of recent ontology research.

As the number of ontologies that are made publicly available and accessible on the Web increases steadily, so does the need for applications to use them. A single ontology is no longer enough to support the tasks envisaged by a distributed environment like the Semantic Web. Multiple ontologies need to be accessed from several applications. Mapping could provide a common layer from which several ontologies could be accessed and hence could exchange information in semantically sound manners.

In this paper, Inductive logic programming (ILP) that is used to structurally map the concepts of different ontologies, aims to solve several problems, especially in machine learning.

This paper proposed a new method to map two ontologies using inductive logic programming. In this method, information about instances transformed to first-order logic predicates, and then ontology mapping is learned from these logic predicates using inductive logic algorithms.

The rest of this paper is organized as follows: section 2 presents a brief description of inductive logic programming, section 3 has a brief overview on the related works. Section 4 introduces the proposed learning-based ontology alignment method. In section 5 some experimental results are presented. Finally, section 6 draws conclusions as well as the future works.

II. A BRIEF INTRODUCTION TO ILP

Inductive logic programming was introduced from early 70th. ILP had gold ages from 1987 to 1996 and in these years was used to solve several problems. ILP is the study of learning methods for data and rules that are represented in first-order predicate logic. Predicate logic allows for quantified variables and relations and can represent concepts that are not expressible using examples described as feature vectors. ILP can be seen as the intersection between logic programming and inductive machine learning.

If-then rules in first-order logic are formally referred to as Horn clauses. A more formal definition of the ILP problem follows [1], [2]:

978-1-5386-5364-7/18/$31.00

- **Given:**
  - Background knowledge, B, a set of Horn clauses.
  - Positive examples, P, a set of Horn clauses (typically ground literals).
  - Negative examples, N, a set of Horn clauses (typically ground literals).
- **Find:** A hypothesis, H, a set of Horn clauses such that:
  - $\forall e \in E^{+} : B \wedge H \rightarrow e$ (Completeness)
  - $\forall e \in E^{-} : B \wedge H \nrightarrow e$ (Consistency)

A. *Generalization with Inverse Resolution*

Inverse resolution is used to produce general clauses from specific examples by inverting the resolution rule of deductive inference. For generalization, inverse substitution is used, which means replacing constants with variables and at the same time ensuring that the original example(s) can be restored by ordinary substitution (Figure 1).

B. *Relative Least General Generalization (rlgg)*

Least general generalization (lgg) of two clauses is such that by applying two substitutions on the same variable in the lgg, two clauses will be regenerated, e.g.

lgg(parent(ann,mary),parent(ann,tom)) = parent(ann,X);

A rlgg is the most specific generalization of two clauses relative to the given background information, e.g.

C_1 = daughter(mary,ann) :- female(mary); parent(ann,mary);

And

C_2 = daughter(eve,tom) :- female(eve); parent(tom,eve);

Then the least general generalization of the two clauses is

lgg(C_1,C_2) = daughter(X,Y) :- female(X); parent(Y,X);

Relative least general generalization is the least general clause more general than both C_1 and C_2 regarding to the background knowledge B.

III. RELATED WORK

This section presents a brief overview of ontology alignment (or ontology matching) techniques such as some algorithms and tools are currently used in this realm of semantic web science.

In this field, many different approaches were proposed for introducing and developing measurements, evaluations and performance improvements on matching systems.

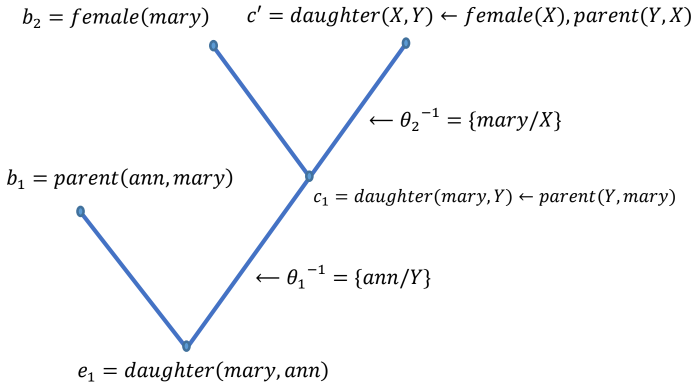


Fig. 1. Inverse resolution

In 2006 and In [3], the authors used inductive logic programming to derive rules from database. In this approach data records transforms to logic predicates and then KDD occurs by logic induction. This work shows that ILP can solve knowledge discovery and rule mining problems effectively.

Ichise proposed an analysis of similarity measures for the ontology mapping problem in 2010 [4]. In order to reach this goal, 48 similarity measures such as string matching and knowledge based similarities that have been widely used in ontology mapping systems were introduced. These similarity measures were investigated by precise analysis on a real-world dataset and As a result, it was possible to identify 22 effective similarity measures for the ontology mapping problem among 48 existing similarity measures in the research. In order to test whether the identified similarity measures are effective for the ontology mapping problem or not, experiments were performed with all 48 similarity measures and the 22 identified similarity measures by using two important and prominent machine learning methods, decision tree and support vector machine.

In 2012, Liu et al. presented an ontology matching approach, which uses multi-strategy mapping technique to explore both linguistic and structural similarity measurements between concepts [5]. Their approach was to combine different similarity measurements called a similarity cube. Cutting these similarity cubes leads to obtain similarity vectors that constitute the similarity space, and then, mapping discovery can be converted into binary classification. Since Support Vector Machine (SVM) can attain best tradeoff between complexity of model and learning capability when solving small samples and the nonlinear problem, the authors decide to employ SVM in their approach.

In ontology matching area and its peripheral tasks, YAM++ and AgreementMakerLight (AML) are two of the best and the most dominant tools for high-performance ontology alignment

978-1-5386-5364-7/18/$31.00 ©2018 IEEE

tasks with approximately satisfiable *F-measure* and accuracy measurements which are web-based and visual application (with user interface) respectively.

So, in [6], the authors attempted to present the capabilities of their ontology matching tool named YAM++. They showed that YAM++ is able to discover mappings (or alignments) between entities (or concepts) of given two different ontologies by using machine learning approaches in addition to demonstrate that if the training datasets are not available, YAM++ can discover mappings by using information retrieval (IR) techniques. Also it is able to deal with multi-lingual ontologies matching problems.

Euzenat and Shvaiko proposed a classification framework for ontology matching [7]. This framework can be traced in both top-down and bottom-up approaches. In top-down trace, matching technique is the classification criteria, and matching methods are classified according to element-based and structure-based, and in next level according to syntax-based and semantic-based. In bottom-up trace, kind of input is the focus of classification.

The complete classification framework is shown in Figure 2.

As you can see in the Figure 2, our approach for ontology matching in order to achieving alignments between concepts, approximately is a hybrid approach according to this following classification.

It is located in *String-based* category which is the syntactic form of *Element-level* in matching techniques because of our lexical similarity measurements and can be located as structural model of content-based matching techniques in *graph-based* category because of the structural analysis on the concepts of the OWL graph structure as nodes by our own ILP tool.

Also, it can be considered strongly in *Instance-based* category which is in structure-level of techniques because of our concentration on positive examples as instances of concepts and considering analysis of relations among them by the tool.

Shvaiko and Euzenat have studied several ontology matching techniques and summarized the state of the art and future challenges of ontology matching [8].

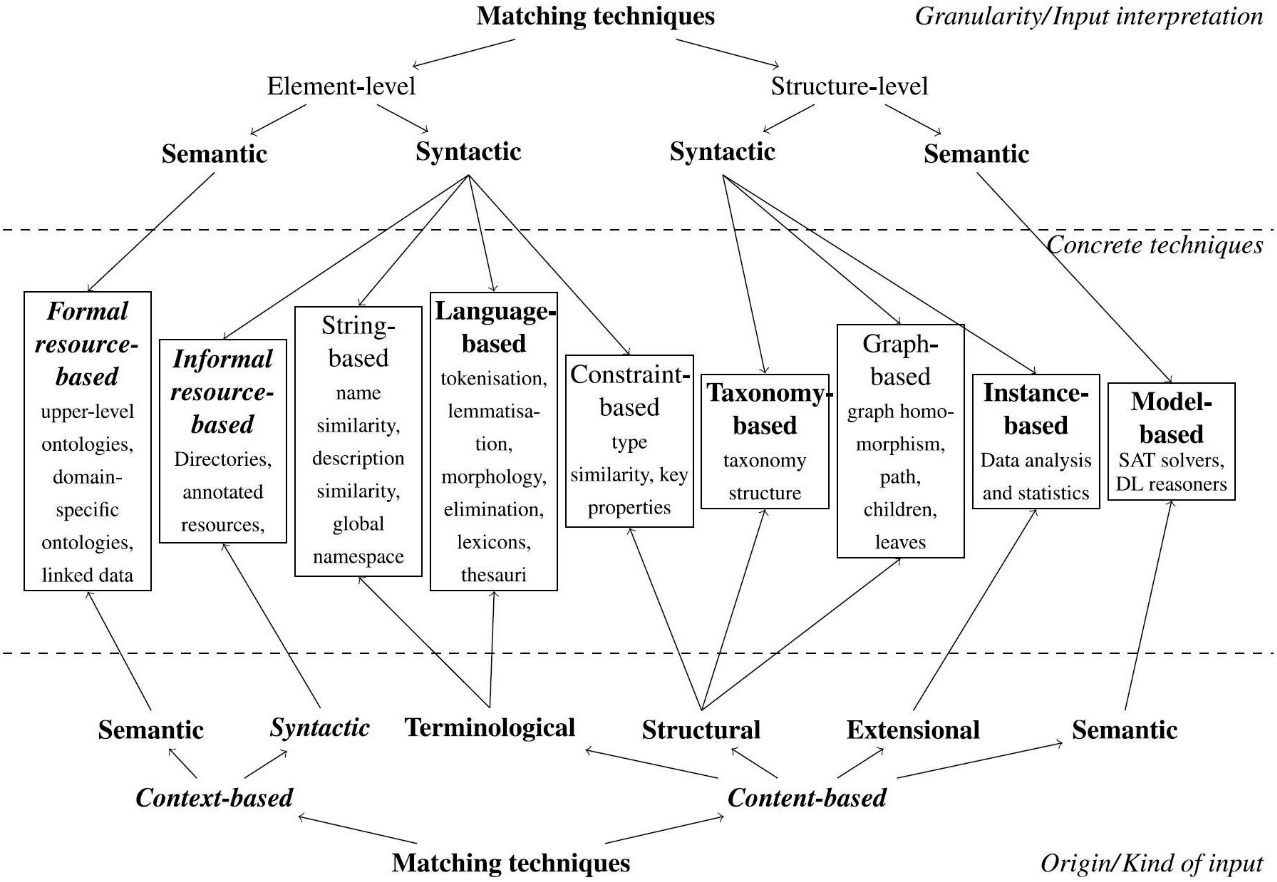


Fig. 2. Ontology matching classification framework [7]

978-1-5386-5364-7/18/$31.00 ©2018 IEEE

They also proposed an analytical comparison between seven systems, including SAMBO, Falcon, DSsim, RiMOM, ASMOV, Anchor-Flood, and AgreementMaker according to several criteria, such as input, output, GUI, operation, terminology, structure, extension and semantic. Also they mentioned 8 challenges in this area: large-scale evaluation, efficiency of ontology matching, matching with background knowledge, matcher selection and self-configuration, user involvement, explanations of ontology matching, collaborative and social ontology matching, and alignment infrastructure.

The other visual and strong tool with remarkable accuracy and significant run-time results to discover alignments between correspondent concepts of two or more different ontologies is AgreementMakerLight (AML) which was introduced in 2013 and it is the advanced version of a tool, AgreementMaker, one of the leading ontology matching systems that is combination of a flexible and extensible framework with a comprehensive user interface which was introduced by Cruz et al. in 2007.

So that in [9], Faria et al. described their new developed core framework, AgreementMakerLight, focused on computational efficiency and designed to discover mappings and handle very large ontologies, while preserving most of the flexibility and extensibility of the original AgreementMaker framework. They evaluated the efficiency of AgreementMakerLight in two OAEI[1] tracks: Anatomy and Large Biomedical Ontologies and after that obtained excellent run-time results.

Also one year later, in 2014, Faria et al. indicated that AgreementMakerLight (AML) is a scalable automated ontology matching system developed primarily for the life sciences domains [10]. It can handle large ontologies efficiently, contains an innovative alignment repair algorithm, and has a graphical user interface which makes it easy to use.

In [11], the authors expressed an approach for matching heterogeneous hierarchical ontologies belong to the same domain through the conceptual interpretation. They had designed rules that can control heterogeneities and inconsistencies which found in hierarchical ontologies. These rules aim to resolve the matching complexities in the hierarchical ontologies.

Also In 2014, Otero-Cerdeira et al. presented a new ontology matching algorithm called OntoPhil which relies on the exploitation of some initial correspondent concepts (i.e. alignments) or binding points that connect the two input ontologies [12]. First of all, it computes these sticky points using a new lexical similarity measurement which combines the information from a terminological matcher with an external one. After that, using the initial binding points as basis and exploiting the specific features of the external structure of the matched ontologies lead to discover new binding points. Finally, the final alignments are obtained by sieving these binding points automatically.

In 2015, Amin et al. presented a performance-based ontology matching system [13]. This system provides effectiveness-independent data parallel ontology matching resolution over parallelism-enabled platforms, since platforms such as desktop and cloud are equipped with parallelism-enabled multicore processors. Their system decomposes complex ontologies into smaller, simpler, and scalable subontologies in terms of the needs of matching algorithms. Matching process over these subontologies is divided to independent matching requests, matching jobs, and matching tasks, running in parallel over parallelism-enabled platforms. Execution of these matching algorithms is aligned for the minimization of the matching space during the matching process.

Brahma and Refoufi [14] proposed an alignment algorithm of two ontologies for the same domain based on different techniques with the use of WordNet as a complementary resource.

Also Otero-Cerdeira et al. performed a literature review in ontology matching [15]. They retrieved articles related to ontology matching that have been published since 2003, to classify and identify research lines relevant for ontology matching.

Also in 2015 and in [16], the authors investigated how the mapping process changes when such a rich conceptualization of the target ontology is available. They developed a translation algorithm that automatically rewrites a mapping from the source schema (or ontology) to the target ontology into an equivalent mapping from the source to the target databases and then, they showed how to handle this problem when an ontology is available also for the source.

In order to provide quick matching and in time processing, a system should control matching precision, runtime complexity and performance issues as a matching strategy.

Thus in [17], Mountasser et al. presented a new hybrid ontology matching approach that first benefit from the opportunities offered by parallel platforms, and then from ontology matching techniques, while applying a resource based decomposition to improve the performance of the system.

Nowadays, in recent years, web researchers investigate on ontology matching and finding alignments (i.e. correspondences) in big data models, because of the role of big data in information economy of tomorrow.

---

[1] Ontology Alignment Evaluation Initiative (http://oaei.ontologymatching.org/2017/anatomy/index.html)

978-1-5386-5364-7/18/$31.00 ©2018 IEEE

For example, in 2017 and in [18], Frimpong aimed to develop robust and comprehensive ontology matching algorithms that can find high-quality alignments between different ontologies while addressing the variety problem associated with Big Data.

Also in [19], Zhang et al. proposed a new hybrid information content (i.e. IC) computing method and a measure of semantic similarity between concepts and their results showed that their measure can improve the similarity accuracy. They evaluated their proposed approach by comparing the correlation coefficients between five similarity measures for correspondent concepts (their approach and four other similarity methods) and the artificial data, so that the results exhibit that their proposal outperforms the previous similarity measures in this area.

In the same year, Cerón-Figueroa et al. proposed a model of pattern classification and its application to align instances from different ontologies [20]. They modeled ontology matching problem as a binary pattern classification.

Also in 2017, Ochieng and Kyanda explored the use of a predictive statistical model to establish an alignment between two different ontologies [21]. They showed how to integrate ontology partitioning and parallelism in the ontology matching process in order to make the statistical predictive model scalable to large ontology matching tasks. Unlike most ontology matching tools which establish one-to-one cardinality mappings, their statistical model generates one-to-many cardinality mappings of concepts from input ontologies.

IV. PROPOSED METHOD TO LEARN ONTOLOGY MAPPING FOR ALIGNMENT

Definition 1: Ontology is defined as a 5-tuple O(C, R, I, IC, RF), where:

- C is a set of concepts
- R is a set of relations (i.e. SubSetOf)
- I is a set of instances
- $IC: I \rightarrow C$, is a function that maps each instance to corresponded concept.
- RF $\subseteq (C \times R \times C)$ is a set of relations between concepts.

Definition 2: Ontology mapping is defined as the task of relating the vocabulary of two ontologies that share the same domain of discourse in such a way that the mathematical structure of ontological signatures and their intended interpretations, as specified by the ontological axioms, are respected [22].

Formally ontology mapping can be defined as a function: $f:(C,R) \rightarrow (C',R')$, which relates signature of two ontologies.

In proposed method, mapping between concepts in order to make some valid alignments is learned from structural similarities between instances using induction. The outline of this method is as follows:

- Generate background knowledge from concept taxonomies (Algorithm 1).
- Add extra background knowledge to our background knowledge (i.e. domain knowledge, expert knowledge and etc.) if exist.
- Generate positive and negative example set from instances.
- Run induction algorithm according to algorithm 2 (using our own tool as an ILP tool that we have implemented before).
- Interpret the result of induction algorithm in previous step as ontology mapping.

In following, some steps are explained in more details.

1) Generate Background Knowledge from Concept Taxonomies

In this step, we have two OWL files of two different ontologies as inputs in order to match corresponding concepts of these ontologies. These OWL files are converted to some logical rules in order to produce our background knowledge B with respect to specifications, properties and special tags exist in tag-based region of each concept or class in each OWL file input (see Table 1).

These produced logical rules as background knowledge of the system have a specific format to be compatible with our ILP tool. We tried to show a general and brief view of how our rule-based background knowledge is produced by two different OWL ontologies in Algorithm 1.

Algorithm 1. A general view to generate background knowledge from two different ontologies.

Algorithm 1: Generating Background Knowledge from Two Different Ontologies

```
Input: Two different ontologies.
$O_1(C_1, R_1, I_1, IC_1, RF_1), O_2(C_2, R_2, I_2, IC_2, RF_2)$
Output: Background knowledge B.
$\forall (c_m, r, c_n) \in RF_1 : B = B \cup \{r_1(c_m, c_n)\}$
$\forall (c_m, r, c_n) \in RF_2 : B = B \cup \{r_2(c_m, c_n)\}$
```

2) *Generating Positive and Negative Examples*

We generated our rule-based positive examples by using RDF file of reference alignment in datasets that is converted to corresponding logical rules with a specific format compatible with our ILP tool. A few percent of this positive example set is considered as a *Train Set* (in our experiments, we used from good and efficient amount of Train Set of about 10%) that is a set of training positive examples used by our ILP tool to learn from it and produce some valid alignments as much as possible.

The remaining percent of this positive example set is considered as a *Test Set* to test and evaluate our proposed method and ILP tool.

The percentage of alignment coverage of these two sets together, i.e. *Train Set* and *Test Set* simultaneously, by our ILP tool can determine the *Recall* measure of this evaluation.

Also, we generated our rule-based negative examples by running our implemented program written in MatLab that produces some of the possible negative examples in order to teach our ILP tool of possible invalid alignments that it should not generate them and try to keep the *Precision* measure up.

3) *Running Inductive Algorithm by Our ILP Tool*

After producing logical predicates from ontologies and instances, inductive logic programming algorithm as Algorithm 2 is applied on these predicates and after that, some general predicates and clauses will be generated, which are correspondence to mappings or alignments between concepts of ontologies.

Finally, the outputs of our ILP tool are some general logical rules that each one can cover some percent of the positive examples (i.e. reference alignments).

It is noteworthy that these coverings might have intersection with each other. Thus, by applying these general produced logic rules on *Train Set* and *Test Set*, we can have our concept alignment results. Also we can evaluate them by specific datasets and criteria and compare them with some related tools in this area.

Although we will see in the following example that lexical similarity of these concepts plays an important role to achieve some kind of valid alignments between concepts. So, we implemented a program written in MatLab to produce some rules related to lexical similarity between pairs of concepts. For this reason, we used a lexical similarity measurement called *Set-Trigram* similarity to apply lexical similarity on anatomical concepts of these ontologies to achieve more valid alignments.

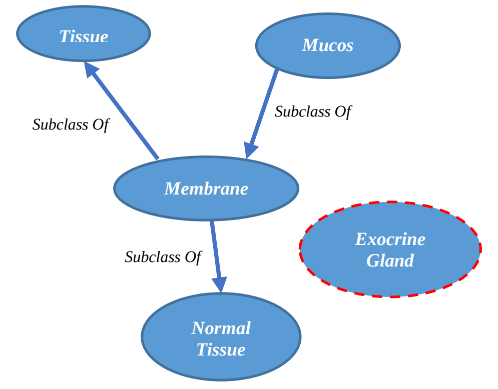
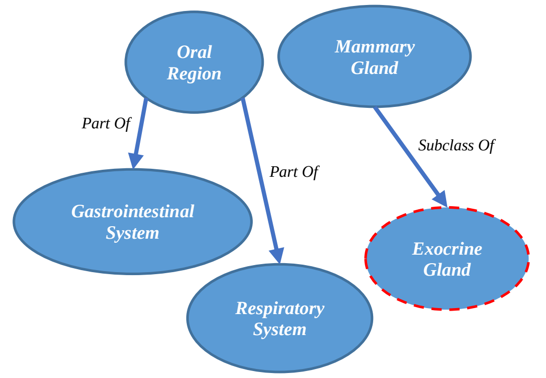
Obviously those rules that are produced by applying this lexical similarity measurement on anatomical concepts, are added to our mentioned background knowledge B to be a part of our ILP tool inputs.

Algorithm 2. The proposed induction algorithm used in our ILP tool

Algorithm 2: The Proposed Induction Algorithm

```
Input: Positive examples PE, Negative examples NE, Background knowledge B.
Output: Set of produced predicates (or rules) H.
Procedure Induction Algorithm:
    $R \longleftarrow \emptyset$;
    Construct a hash map of all constants and containing clauses in B;
    foreach clause $c \in PE$ do
        if $c$ is not covered yet then
            make a predicate P as c:- ;
            foreach predicate $P' \in Induce(P)$ do
                $H \longleftarrow H \cup P'$;
            end foreach
        end if
    end foreach
    Remove duplicate predicates from H;
    Remove weak predicates from H;
    Remove predicates which produce a negative example from H;
```

TABLE I. CONVERSION OF THE OWL NOTATION OF CONCEPTS IN THE ANATOMY ONTOLOGIES

	A Part of Human Anatomy	***A Part of Mouse Anatomy***
Graph and Dependencies of Concepts		
OWL Notation of A Concept	<owl:Class rdf:about="Human.owl#NCI_C12234"> <rdfs:label rdf:datatype="XMLSchema#string"> sublingual salivary gland </rdfs:label> <rdfs:subClassOf rdf:resource="Human.owl#NCI_C12999"/> <oboInOwl:hasRelatedSynonym rdf:resource="Human.owl#genid4893"/> <oboInOwl:hasRelatedSynonym rdf:resource="Human.owl#genid4894"/> </owl:Class>	<owl:Class rdf:about="Mouse.owl#MA_0001588"> <rdfs:label rdf:datatype="XMLSchema#string"> sublingual gland </rdfs:label> <rdfs:subClassOf rdf:resource="Mouse.owl#MA_0002478"/> </owl:Class>
Logical Rules of A Concept	Label(NCI_C12234,sublingual salivary gland); SubClassOf(NCI_C12234,NCI_C12999); RelatedSynonym(NCI_C12234,genid4893); RelatedSynonym(NCI_C12234,genid4894);	Label(MA_0001588,sublingual gland); SubClassOf(MA_0001588,MA_0002478);

4) *An Example*

In this example, we used a pair of concepts from the anatomy datasets consist of the Adult Mouse Anatomy (2,744 classes) and the NCI Thesaurus (3,304 classes) describing the human anatomy that are illustrated in next sections. We show how to apply proposed method to find ontology mapping between correspondences of concepts in the different ontologies.

By applying similarity assessment, we find that the following terms are similar:

Human Anatomy (NCI_C)	Mouse Anatomy (MA)
sublingual salivary gland	*sublingual gland*

In next step, by applying algorithm 1, following background knowledge are created:

Label(NCI_C12234,sublingual salivary gland)
RelatedSynonym(NCI_C12234,genid4894)
Label(genid4894,sublingual gland)
Label(MA_0001588,sublingual gland)

After generating positive and negative examples that showing the correct and incorrect alignments respectively, then by using Algorithm 2 and using our own ILP tool in order to find some valid alignments between pairs of concepts, the result of applying logic induction is illustrated in TABLE II and TABLE III that shows the progress of producing general and logical alignment rules by positive examples of *Train Set* and

978-1-5386-5364-7/18/$31.00 ©2018 IEEE

TABLE II. PRODUCING ALIGN RULES BY POSITIVE EXAMPLES OF TRAIN SET

Step	***Base Predicate***	***Augmented Predicate***	***Substitution***	***Produced Rule***
1	Align(NCI_C12234,MA_0001588)	Label(MA_0001588,sublingual gland)	MA_0001588/X1	Align(NCI_C12234,X1) :- Label(X1,sublingual gland)
2	Align(NCI_C12234,X1) :- Label(X1,sublingual gland)	Label(genid4894,sublingual gland)	Sublingual gland/X2	Align(NCI_C12234,X1) :- Label(X1,X2), Label(genid4894,X2)
3	Align(NCI_C12234,X1) :- Label(X1,X2), Label(genid4894,X2)	RelatedSynonym(NCI_C12234, genid4894)	genid4894/X3 NCI_C12234/X4	Align(X4,X1) :- Label(X1,X2), Label(X3,X2), RelatedSynonym(X4,X3)

TABLE III. PRODUCING CONCEPT ALIGNMENTS BY LOGIC ALIGN RULES

Step	***Produced Rule***	***Background Knowledge***	***Substitution***	***Align Progress***
1	Align(X4,X1) :- Label(X1,X2), Label(X3,X2), RelatedSynonym(X4,X3)	RelatedSynonym(NCI_C12234,genid4894)	X4/NCI_C12234 X3/genid4894	Align(NCI_C12234,X1) :- Label(X1,X2), Label(genid4894,X2)
2	Align(NCI_C12234,X1) :- Label(X1,X2), Label(genid4894,X2)	Label(genid4894,sublingual gland)	X2/sublingual gland	Align(NCI_C12234,X1) :- Label(X1,sublingual gland)
3	Align(NCI_C12234,X1) :- Label(X1,sublingual gland)	Label(MA_0001588,sublingual gland)	X1/MA_0001588	Align(NCI_C12234,MA_0001588)

shows the progress of producing some alignments between concepts by using align rules respectively.

The algorithm discovers that *sublingual salivary gland* concept in Human Anatomy ontology is structurally similar to *sublingual gland* in Mouse Anatomy ontology as well as their lexical similarity with score 0.846. This structural similarity is discovered by analyzing instances and their locations in their taxonomies.

V. EXPERIMENTAL RESULTS

We have implemented a tool for this algorithm that is written in C#. We have evaluated the proposed method on anatomy domains whose characteristics are shown in TABLE IV. These domains describes body organs and systems in Human and Mouse anatomies which have 3304 and 2744 nodes respectively.

These ontologies are also used in many matching systems and tools to evaluate the quality of their alignments and the performance of their alignment systems.

A. *DataSets Used in Experiments*

The anatomy real world case is about matching the Adult Mouse Anatomy (2,744 classes) and the NCI Thesaurus (3,304 classes) describing the Human Anatomy.

Our task consists of finding some valid alignments between the Adult Mouse Anatomy and a part of the NCI Thesaurus (describing the human anatomy). Besides their large size and a conceptualization that is only to a limited degree based on the use of natural language, they also differ from other ontologies with respect to the use of specific annotations and roles, e.g. the extensive use of the *part_of* relation.

1) *NCI Thesaurus: Human Anatomy*

This thesaurus published by the National Cancer Institute (NCI) and contains the working terminology of many data systems in use at NCI [23]. Among its 37,386 concepts, 3,304 (8.8%) of them are anatomical entities (*anatomic structure, system, or substance* hierarchy). For example, the concept *liver* is identified by *NCI_C12392* and has several synonyms (e.g., *hepatic organ system*).

Additionally, *liver* is subsumed by *organ* and related to *abdominal cavity* (*has_property*) and to *gastrointestinal system* (*part_of*).

2) Adult Mouse Anatomical Dictionary

The Adult Mouse Anatomical Dictionary has been developed as part of the mouse Gene Expression Database (GXD) project to provide standardized nomenclature for anatomical concepts in the adult mouse [24]. It will be used to name and collect different types of data pertinent to anatomy, such as gene expression patterns and phenotype information, which will contribute to an integrated description of biological entities in the mouse.

The ontology contains more than 2,400 unique terms, is structured as a DAG[2] and is organized hierarchically in both spatial and functional ways. For example, the concept *liver* is identified by *MA_0000358* and is a *child* of *(is_a) abdomen organ* as well as *part_of* the *liver/biliary system*. Of the 3,304 NCI Thesaurus terms, over 1,800 terms correspond to entities that are not included in the Adult Mouse anatomical Dictionary, such as cell types and subcellular components. Thus, only about 1,500 would be expected to be candidates for matching.

B. Evaluations

As a qualified and known assessment approach for evaluate and validate ontology alignment (matching) tools or systems, there are some measurements called *Precision*, *Recall* and *F-Measure*. The results (Figure 3) show that AML achieves a good *Precision* about 95% and a remarkable *Recall* about 93.6%, and so with respect to its *F-Measure* about 94.3%, this tool was one of the best tools with the highest *F-Measure* in OAEI 2017 (see Figure 3 and Figure 4).

TABLE IV. CHARACTERISTICS OF ONTOLOGIES OF HUMAN AND MOUSE ANATOMIES

Ontologies	***Human Anatomy***	***Mouse Anatomy***
#Classes (Concepts or Nodes)	**3,304**	**2,744**
#Roots	**7**	**8**
#Leaf Nodes	**2,631**	**2,261**
#Max Depth	**13**	**8**
#Manual Alignments (#Mappings)	**1,516**	

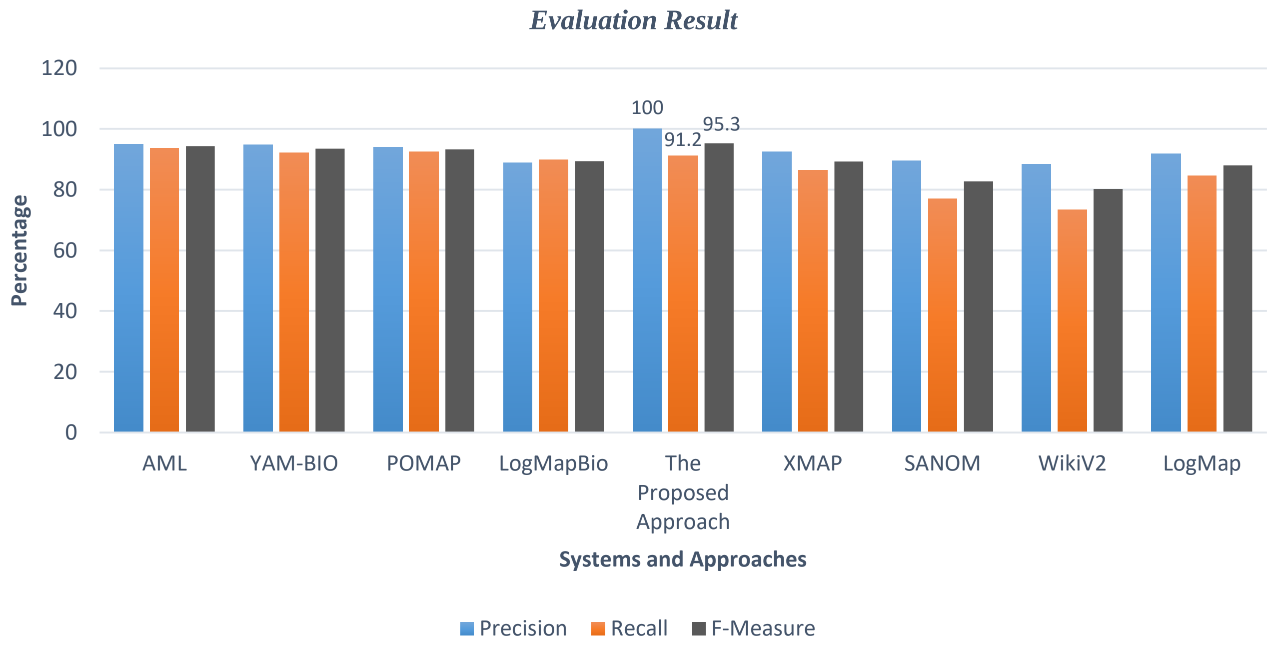


Fig. 3. Evaluation result

---

[2] Directed Acyclic Graph

978-1-5386-5364-7/18/$31.00 ©2018 IEEE

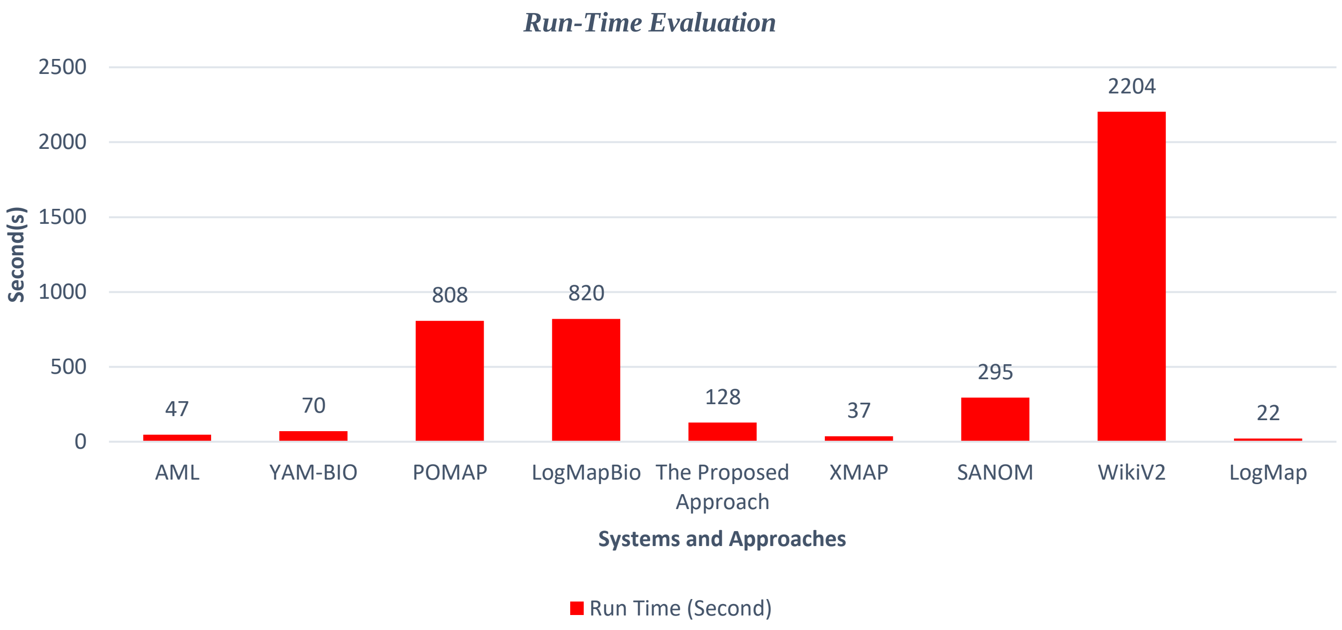
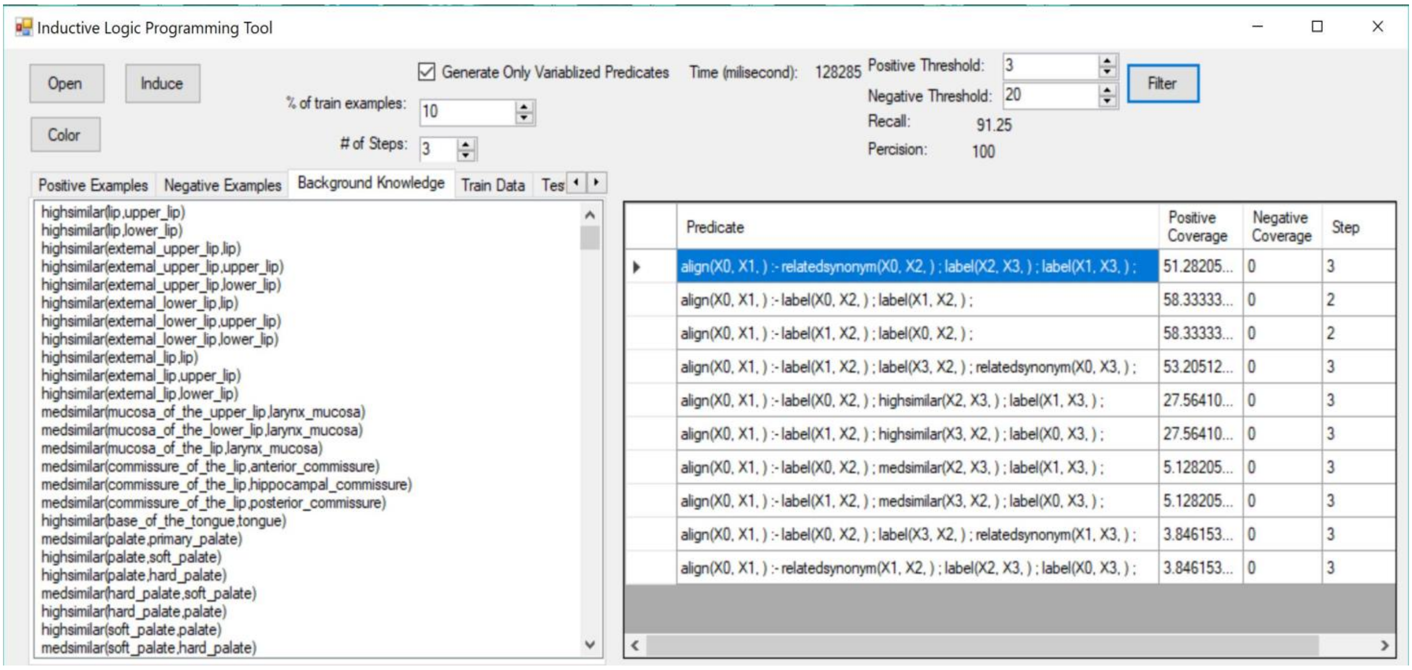


Fig. 4. Run-time result



☑ Generate Only Variablized Predicates

% of train examples: 10

\# of Steps: 3

Time (milisecond): 128285

Positive Threshold: 3

Negative Threshold: 20

Recall: 91.25

Percision: 100

```
highsimilar(lip,upper_lip)
highsimilar(lip,lower_lip)
highsimilar(external_upper_lip,lip)
highsimilar(external_upper_lip,upper_lip)
highsimilar(external_upper_lip,lower_lip)
highsimilar(external_lower_lip,lip)
highsimilar(external_lower_lip,upper_lip)
highsimilar(external_lower_lip,lower_lip)
highsimilar(external_lip,lip)
highsimilar(external_lip,upper_lip)
highsimilar(external_lip,lower_lip)
medsimilar(mucosa_of_the_upper_lip,larynx_mucosa)
medsimilar(mucosa_of_the_lower_lip,larynx_mucosa)
medsimilar(mucosa_of_the_lip,larynx_mucosa)
medsimilar(commissure_of_the_lip,anterior_commissure)
medsimilar(commissure_of_the_lip,hippocampal_commissure)
medsimilar(commissure_of_the_lip,posterior_commissure)
highsimilar(base_of_the_tongue,tongue)
medsimilar(palate,primary_palate)
highsimilar(palate,soft_palate)
highsimilar(palate,hard_palate)
medsimilar(hard_palate,soft_palate)
highsimilar(hard_palate,palate)
highsimilar(soft_palate,palate)
medsimilar(soft_palate,hard_palate)
```

	Predicate	Positive Coverage	Negative Coverage	Step
▶	align(X0, X1,) :- relatedsynonym(X0, X2,) ; label(X2, X3,) ; label(X1, X3,) ;	51.28205...	0	3
	align(X0, X1,) :- label(X0, X2,) ; label(X1, X2,) ;	58.33333...	0	2
	align(X0, X1,) :- label(X1, X2,) ; label(X0, X2,) ;	58.33333...	0	2
	align(X0, X1,) :- label(X1, X2,) ; label(X3, X2,) ; relatedsynonym(X0, X3,) ;	53.20512...	0	3
	align(X0, X1,) :- label(X0, X2,) ; highsimilar(X2, X3,) ; label(X1, X3,) ;	27.56410...	0	3
	align(X0, X1,) :- label(X1, X2,) ; highsimilar(X3, X2,) ; label(X0, X3,) ;	27.56410...	0	3
	align(X0, X1,) :- label(X0, X2,) ; medsimilar(X2, X3,) ; label(X1, X3,) ;	5.128205...	0	3
	align(X0, X1,) :- label(X1, X2,) ; medsimilar(X3, X2,) ; label(X0, X3,) ;	5.128205...	0	3
	align(X0, X1,) :- label(X0, X2,) ; label(X3, X2,) ; relatedsynonym(X1, X3,) ;	3.846153...	0	3
	align(X0, X1,) :- relatedsynonym(X1, X2,) ; label(X2, X3,) ; label(X0, X3,) ;	3.846153...	0	3

Fig. 5. ILP tool result

978-1-5386-5364-7/18/$31.00 ©2018 IEEE

Our proposed method achieves the highest *Precision* 100% in similar case and has 91.2% for *Recall* measurement (see Figure 5). Thus, our approach's *F-Measure* is about 95.3% which is the highest amount and a very good and efficient one among all of the other tools on similar datasets. So, as a future work, it needs to some manipulations in order to optimize and rebalance its *Precision* and *Recall* measure and try to improve its performance by reducing its run-time. A question appears here, why this method could find all mappings that can be find by human? It may be that man use induction and similar approach to find mappings. It may be that ILP is a simulation of human brain learning mechanism used in this context.

VI. CONCLUSION

This paper proposed a new approach to find ontology mapping using inductive logic programming. In this approach, each concept and relation in ontologies are transformed to logic predicates. Also, each instance is transformed to Horn clause, and then applying logic induction, some general predicates will be generated. These predicates are corresponding to ontology mappings. This approach can resolve structural inconsistencies between two ontologies. It is based on well-known and verified theory (Logic). One of advantages of this approach is the ability to use background knowledge, as an input to induction algorithm. It is very flexible and powerful approach, and can find all existing mappings theoretically.

Some drawbacks of this approach are its dependency on common instances. If two ontologies have not enough common instances, this approach can't work properly. Finally, in most cases, induction is time consuming task, so this approach needs some optimizations to apply in practical and real world problems.

ACKNOWLEDGMENT

This research was partially supported by University of Tehran.

REFERENCES

[1] N. Lavrač and S. Džeroski, "Inductive logic programming : techniques and applications", E. Horwood, 1994.

[2] S. Muggleton, "Inverse entailement and progol", *New Gener. Comput.*, vol. 13, no. 3–4, pp. 245–286, Dec. 1995.

[3] A. Kamandi, S. Jalili, and J. Habibi, "Knowledge Discovery in Database Using Inductive Logic Programming", CSICC06, 901-905, 2006 (in Persian).

[4] R. Ichise, "An analysis of multiple similarity measures for ontology mapping problem", *Int. J. Semant. Comput.*, vol. 4, no. 1, pp. 103–122, Mar. 2010.

[5] L. Liu, F. Yang, P. Zhang, J.-Y. Wu, and L. Hu, "SVM-based ontology matching approach", *Int. J. Autom. Comput.*, vol. 9, no. 3, pp. 306–314, Jun. 2012.

[6] D. Ngo and Z. Bellahsene, "YAM++ : A Multi-strategy Based Approach for Ontology Matching Task", 2012, pp. 421–425.

[7] J. Euzenat and P. Shvaiko, *Ontology matching*. 2013.

[8] P. Shvaiko and J. Euzenat, "Ontology Matching: State of the Art and Future Challenges", *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 158–176, Jan. 2013.

[9] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto, "The AgreementMakerLight Ontology Matching System", 2013, pp. 527–541.

[10] D. Faria, C. Pesquita, E. Santos, I. Cruz, and F. Couto, "AgreementMakerLight: A Scalable Automated Ontology Matching System", *DILS 2014*, 2014.

[11] S. Khan and M. Safyan, "Semantic matching in hierarchical ontologies", *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 26, no. 3, pp. 247–257, Sep. 2014.

[12] L. Otero-Cerdeira, F. J. Rodríguez-Martínez, T. Valencia-Requejo, and A. Gómez-Rodríguez, "OntoPhil: Exploitation of binding points for ontology matching", 2014, pp. 5–15.

[13] M. B. Amin, W. A. Khan, S. Lee, and B. H. Kang, "Performance-based ontology matching", *Appl. Intell.*, vol. 43, no. 2, pp. 356–385, Sep. 2015.

[14] B. Brahma and A. Refoufi, "Ontology Matching Algorithms", in *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication - IPAC '15*, 2015, pp. 1–5.

[15] L. Otero-Cerdeira, F. J. Rodríguez-Martínez, and A. Gómez-Rodríguez, "Ontology matching: A literature review", *Expert Syst. Appl.*, vol. 42, no. 2, pp. 949–971, Feb. 2015.

[16] G. Mecca, G. Rull, D. Santoro, and E. Teniente, "Ontology-based mappings", *Data Knowl. Eng.*, vol. 98, pp. 8–29, Jul. 2015.

[17] I. Mountasser, B. Ouhbi, and B. Frikh, "Hybrid large-scale ontology matching strategy on big data environment", in *Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services - iiWAS '16*, 2016, pp. 282–287.

[18] R. A. Frimpong, “Ontology Matching Algorithms for Data Model Alignment in Big Data”, 2017, pp. 195–204.

[19] X. Zhang, S. Sun, and K. Zhang, “A Novel Comprehensive Approach for Estimating Concept Semantic Similarity in WordNet”, Mar. 2017.

[20] S. Cerón-Figueroa *et al.*, “Instance-based ontology matching for e-learning material using an associative pattern classifier”, *Comput. Human Behav.*, vol. 69, pp. 218–225, Apr. 2017.

[21] P. Ochieng and S. Kyanda, “A statistically-based ontology matching tool”, *Distrib. Parallel Databases*, Sep. 2017.

[22] Y. Kalfoglou and M. Schorlemmer, “Ontology mapping: the state of the art”, *Knowl. Eng. Rev.*, vol. 18, no. 1, p. S0269888903000651, Jan. 2003.

[23] S. de Coronado, M. W. Haber, N. Sioutos, M. S. Tuttle, and L. W. Wright, “NCI Thesaurus: using science-based terminology to integrate cancer research results”, *Stud. Health Technol. Inform.*, vol. 107, no. Pt 1, pp. 33–7, 2004.

[24] T. F. Hayamizu, M. Mangan, J. P. Corradi, J. A. Kadin, and M. Ringwald, “The Adult Mouse Anatomical Dictionary: a tool for annotating and integrating data”, *Genome Biol.*, vol. 6, no. 3, p. R29, Feb. 2005.

978-1-5386-5364-7/18/$31.00 ©2018 IEEE