Proceedings of the Asia Pacific Industrial Engineering & Management Systems Conference 2012
V. Kachitvichyanukul, H.T. Luong, and R. Pitakaso Eds.

Ontology matching by using ConceptNet

Mehdi Keshavarz †
Department of Industrial and Systems Engineering
Dongguk University-Seoul, 26, Pil-dong 3-ga, Jung-gu, Seoul, 100-715, South Korea.
Email: Keshavarzm2005@gmail.com

Yong-Han Lee
Department of Industrial and Systems Engineering
Dongguk University-Seoul, 26, Pil-dong 3-ga, Jung-gu, Seoul, 100-715, South Korea.
Email: yonghan@dgu.edu

Abstract. Ontology matching is a main step for integrating overlapping domains of knowledge and establishing interoperation among semantic web application. As information sources grow rapidly, manual ontology matching becomes more tedious and time-consuming and consequently leads to errors and frustration. In this paper we developed the new lexical and semantic similarity measure by using the lexical database ConceptNet. The proposed strategy used new lexical and semantic matching for finding the correspondence entities. In the semantic approach we use the electronic lexical database, ConceptNet for identifying the similar entities and create similarity matrices according to that. We evaluate the proposed measure using standard methods of precision and recall, tested on a well- known benchmark and also compared to other algorithms presented in the paper. The experimental results show the proposed algorithm is effective and outperforms other algorithms.

Keywords: Ontology matching, ConceptNet similarity, Similarity measure, Information Technology, Knowledge and Information Management

1. INTRODUCTION

These days Ontology is a main part of many applications. By using ontology, both the user and the system can communicate with each other using a common understanding of a domain (Abulaish. 2009). Ontology matching is the basis of the interoperation of heterogeneous ontologies in semantic web (Euzenat and Shvaiko. 2007), and is the most effective way to solve the problem of ontology heterogeneity in distributed environment. Also one of the most advantage of ontology is the ability for knowledge sharing among different agents or applications, since multi agent system can be use for solving or implementing complex systems and in MAS each agent in order to communicate with each other they should be a way to understand each other so the best choice is by using ontology. The main task in agent communication is interoperability problem and in order to obtain semantic interoperability in distributed multi-agent Systems, agents need to agree on the basis of different knowledge or ontologies. Ontology is used throughout the multi-agent system to assist the interactions among different agents as well as to improve the quality of the service provided by each agent. The main task of ontology matching, and mapping extraction is to find whether there exists similarities between source ontology and target ontology. This paper, divide into three parts. At first part we extract the information from all ontologies and classify them into corresponding matrices (Birkoff. 1976). We use four matrices, one for classes, one for data properties, one for object properties and one for instances of given ontologies. In the second part we developed the new algorithm for lexical matching between elements of each ontologies and use Levenshtein distance string matching and use stemming algorithm like Paice/Hush (Price.1990) stemmer for comparing strings with each other. Finally in the last part we used the semantic matching by using the ConceptNet for analyzing and checking the similarity matrices. In previous works that has be done in ontology matching area, mostly the use

† : Corresponding Author

WordNet(http://wordnet.princeton.edu/) for checking the semantic similarity of elements, however using WordNet was not so accurate and they should use structural matching for checking the structure of each classes of given ontologies with each other.

2. RELATED WORK

2.1. Ontology

Although there are different definitions for ontology but the best definitions were defined by Gruber and Studer. Gruber (1993) defines ontology as “a specification of a conceptualization”. Another definition is given in (Studer et al, 1998): “ontology is a formal, explicit specification of a shared conceptualization”. By using specification of the conceptualization it means that each ontology consists of the objects, concepts (classes) and other entities that are in the same particular domain and the relationships that hold among them. “Explicit” means that objects, concepts, and other entities are explicitly defined. “Formal” implies that the ontology should be understood by machine or in other words machine readable. “Shared” means that the ontology captures consensual knowledge and is agreed-upon by a group, not just an individual. In generally, the main structure of ontology model consists of three main categories: concept or class, property (data property), and relation (object property). Both ontology and agent technologies are essential to the semantic web, and their combined use will enable the sharing of heterogeneous, autonomous knowledge sources in a capable, adaptable and extensible manner. Ontology uses in different area,for instance Ontology (Chandrasekaran, Josephson, & Benjamins, 1999) is one theory in philosophy that primarily explores the knowledge characteristics of life and real objects. In the artificial intelligence field, it was used to define the content of domain knowledge, express knowledge, and solves communication and commonly shared problems. In the information technology field, it offers much assistance for research and development of e-commerce and knowledge management (Hsu, 2006, Yang, Ding, & Ho, 2003).

2.2. Ontology Mapping

Recently, interoperability and reusability of ontologies with each other is more considered, since the problem of semantic web and most of search engines are finding appropriate information since there exists huge amount of information on internet but there is no efficient mapping systems between these information or documents. It means that similar objects which are described in different ontology structures could be integrated into a new ontology structure and they could be utilized in a particular system. This technology is known as an ontology mapping. As described in (Laurel et al, 2004), there are two types of ontology mapping: source-based and instance-based. Examples of source-based mapping tools are PROMPT (Natalya Fridman Noy and Mark A. Musen, 2000), Chimaera(D. McGuinness *et al.* 2000), and ONIONS(Aldo Gangemi *et al.* 2007) and examples of instance-based mapping tools(Jérôme Euzenat, Pavel Shvaiko. 2007) are FCA-Merge(B. Ganter, R. Wille. 1997) and GLUE(Jérôme Euzenat, Pavel Shvaiko. 2007). Beyond, a new methodology for merging the heterogeneous domain ontologies based on the ConceptNet(http://web.media.mit.edu/~hugo/conceptnet/) and WordNet(http://wordnet.princeton.edu/) which is used as a dictionary to give relationships between concepts detailed in. Ontology is a basis for enabling interoperability across heterogeneous systems, services and applications. One of the most challenging tasks in the area is ontology matching and alignment. However, many methods (Keshavarz and Lee. 2011, Rung-Ching *et al.* 2011, Mikalai Yatskevich and Fausto Giunchiglia.2006, Ismail Akbari and Mohammad Fathian. 2010) and tools(D. McGuinness *et al.* 2000, Natalya Fridman Noy and Mark A. Musen. 2000, Aldo Gangemi et al. 2007, Jérôme Euzenat, Pavel Shvaiko. 2007) have been proposed for finding corresponding entities between ontologies.In this part we review some of the approaches that have been proposed (Euzenat and Shvaiko, 2007) present a comprehensive review of current approaches, classifying them along three main dimensions: granularity, input interpretation, and kind of input. The granularity dimension distinguishes between element-level and structure-level techniques (Euzenat and Shvaiko, 2007). The input interpretation dimension is divided into syntactic, which uses solely the structure of the ontologies; external, which exploits auxiliary resources outside of the ontologies; and semantic, which uses some form of formal semantics to justify results. The kind of input dimension categorizes techniques as terminological, which works on textual strings; structural, which deals with the structure of the ontologies; extensional, which analyzes the data instances; and semantic, which makes use of the underlying semantic interpretation of ontologies.

Most of the work on ontology matching has focused on lexical and semantic or structural approaches. Many of these approaches usually use a common knowledge or a domain-specific thesaurus like WordNet to match words based on linguistic relations between them (e.g. synonyms, hyponyms). In this case, the names of ontology entities are considered as words of a natural language. In overall, the field of domain ontology mapping and alignment includes the following six strategies: (1) Strategies based on lexical matching: They will do integration according to the string based methods for identifying similar entities in given ontologies. (2) Strategies based on linguistic and semantic matching: They complete an integration task according to the linguistic meaning of the words. (3) Structure-based techniques: They consider the ontology entities or their instances to compare their relations with other entities or their instances. (4)Constraint-based strategy: They complete the integrative task according to the constraints in each concept (5) Instance-based matching strategies: they crucially depend on measuring the similarity between sets of annotated instances. (6) Graph-based methods: They treat ontologies as graphs and compare the corresponding sub graphs. So we split these strategies into 4 categories: lexical, semantic, structural and combinatorial (Ismail Akbari and Mohammad Fathian, 2010),

Lexical approaches are string-based methods for comparing string entities in given ontologies. They can be applied to the name of classes or URIs, properties and instances for identifying identical classes, data and object properties and instances in the source and target ontologies based on the similarity of their label or description. These techniques tokenize the strings. So they are typically based on the following intuition: the more similar the strings, the more likely they denote the same concepts. Some sorts of string-based techniques which are extensively used in matching systems are DamerauLevenshtein Distance, edit distance, and n-gram.

DamerauLevenshtein(J.L. Dawson.1974) Distance is an algorithm for calculating distance between strings by Damerau and Levenshtein. It lets you find matches based on proximity, which allows for human errors such as typos and spelling mistakes. For example "tihs" is 1 distance away from "this" so it's safe to assume the latter is implied, conversely "Toralf" and "Titan" are very far so it can be ignored. Edit distance between two objects is the minimal cost of operations to be applied to one of the objects in order to obtain the other one. Edit distances were designed for measuring similarity between strings that may contain spelling mistakes. The edit distance is a dissimilarity $\delta : S \times S \rightarrow [0\ 1]$ where $\delta(s, t)$, is the cost of the less costly sequence of operations which transforms s into t.

$$sim_{EditDistance}(s,t) = 1 - \frac{d}{max(ls,lt)},$$

where:

- *s* and *t*: the two values to compare,
- *d*: the distance (cost) between *s* and *t*,
- *ls* and *lt*: the length of *s* and *t* respectively,
- *max(ls,lt)*: the maximum length between *s* and *t*.

For example, the Anchor Prompt (Jérôme Euzenat, Pavel Shvaiko. 2007) and MLMA+ matcher uses Levenshtein distance as an edit distance to compute the lexical similarity between two entities. N-gram-based approaches take two strings as input and it computes the number of common n-grams, i.e., sequences of n characters, between them. For instance, 3-grams for the string article are: art, rti, tic, icl, cle. Thus, for example, the similarity between article and aricle would be 2/4 = .5, while between article and paper would be 0, and, finally, between article and particle would be 5/6 = .83. It is possible, to add extra characters at the beginning and end of strings for dealing with too small strings.

In semantic analysis strategies usually one or more linguistic resources such as thesauri and Terminologies are used to identify synonym and hyponyms entities. One the electrical lexical database for English and other languages is WordNet. WordNet was designed to establish the connections between four types of Parts of Speech (POS) - noun, verb, adjective, and adverb. The smallest unit in a WordNet is synset, which represents a specific meaning of a word. It includes the word, its explanation, and its synonyms. The specific meaning of one word under one type of POS is called a sense. Each sense of a word is in a different synset. Synsets are equivalent to senses = structures containing sets of terms with synonymous meanings. Each synset has a gloss that defines the concept it represents. For example, the words night, nighttime, and dark constitute a single synset that has the following gloss: the time after sunset and before sunrise while it is dark outside. Synsets are connected to one another through explicit semantic relations. Some of these relations (hypernym, hyponym for nouns, and hypernym and troponym for verbs) constitute is-a-kind-of (homonymy) and is-a-part-of (metonymy for nouns) hierarchies. For example, tree is a kind of plant, tree is a hyponym of plant, and plant is a hyponym of tree. Analogously, trunk is a part of a tree, and we have trunk as a metonyms of tree, and tree is a holonym of trunk. For one word and one type of POS, if there is more than one sense, WordNet organizes them in the order of the most frequently used to the least frequently used (Semcor).Also there are some other tools which can be used in java for semantic analysis which are: OpenNLP (version 1.3.0) is a framework for linguistic analysis including, for instance, components for determining the lexical categories of words (e.g., adjective).

MorphAdorner, is a text processing framework which amongst other components provides means for morphological analysis and generation, i.e., inflection of words.

LexParser also known as the Stanford Parser is a syntactic parser which can be used to determine the grammatical structure of phrases (e.g., noun phrases such as “accepted paper”) or sentences.

Structural approaches can be compared instead of or in addition to comparing the entities or identifier names. These approaches identify similar classes by looking at their relationships to the other classes and also their properties. So two classes of the given ontologies are similar if and only if they have similar or same neighbors and same attributes. For example, Taxonomic structure is a structural methodology which is based on counting the number of edges in the taxonomy between two classes. The structural topological dissimilarity on a hierarchy (Valtchev and Euzenat, 1997) follows the graph distance, i.e., the shortest path distance in a graph taken here as the transitive reduction of the hierarchy. Anchor Prompt tries to find relationships between entities based on the primary relationships recognized before. The central observation behind Anchor-Prompt is that if two pairs of terms from the source ontologies are similar and there are paths connecting the terms, then the elements in those paths are often similar as well. PROMPT consists of an interactive ontology merging tool (Natalya Fridman Noy and Mark A. Musen, 2000) and a graph based mapping dubbed Anchor-PROMPT (Natalya F. Noy and Mark A. Musen,2006). It uses linguistic "anchors" as a starting point and analyzes these anchors in terms of the structure of the ontologies. GLUE (D. McGuinness *et al* 2000) discovers mappings through multiple learners that analyze the taxonomy and the information within concept instances of ontologies. COMA (Shvaiko, P., Euzenat, J. 2005) uses parallel composition of multiple element- and structure-level matchers. Corpus-based matching uses domain-specific knowledge in the form of an external corpus of mappings which evolves over time.

2.3. WordNet (Lexical database)

WordNet (http://wordnet.princeton.edu/) is an electronic lexical database for English (and other languages) created and maintained by Princeton University by Professor George A. Miller since 1985. It is a database of words, primarily nouns, verbs and adjectives, organized into discrete 'senses', and linked by a small set of semantic relations such as the synonym relation and 'is-a' hierarchical relations contains roughly 200 000 word 'senses' (a sense is a 'distinct' meaning that a word can assume). One of the reasons for its success and wide adoption is its ease of use. The upper ontology SUMO (Suggested Upper Merged Ontology) defines the conceptual mapping with respect to WordNet synsets (Mikalai Yatskevich and Fausto Giunchiglia. 2006).WordNet superficially resembles a thesaurus, in that it groups words together based on their meanings. However, there are some important distinctions. First, WordNet interlinks not just word forms, strings of letters, but specific senses of words. As a result, words that are found in close proximity to one another in the network are semantically disambiguated. Second, WordNet labels the semantic relations among words, whereas the groupings of words in a thesaurus do not follow any explicit pattern other than meaning similarity. As can be seen from figure 1, words are represented by resources of type wn:Word@@ref.
The properties wn:hasWordForm and wn:hasLanguage, relate a word to its word form and language respectively. The property wn:hasSense relates words to their senses. WordNet stores information in flat files called lexographer files, which are then processed by a tool called grind to produce the database. WordNet infers the root form of a word at run time, and it is not directly stored in the database. Although grind and the lexicographer files are freely available, modifying and maintaining the database is said to be difficult.

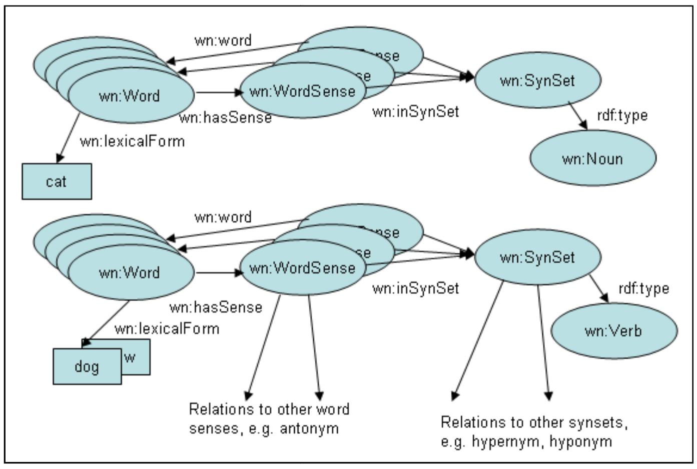


Figure 1: Representation of words by resources

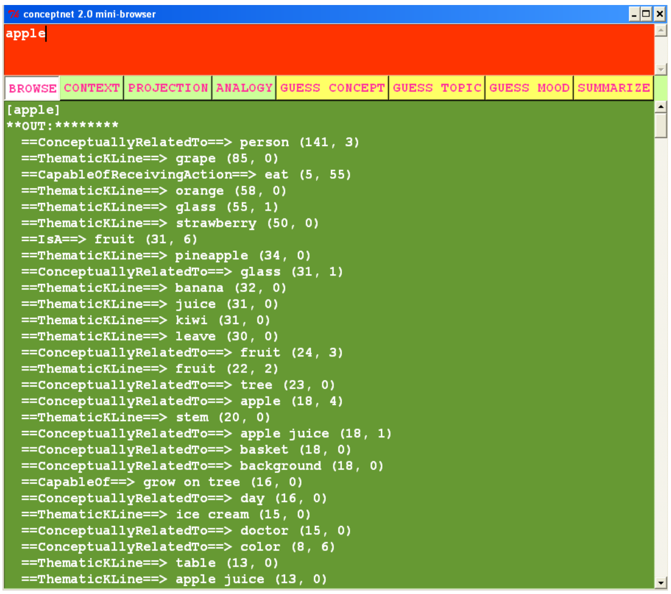
Y is a hypernym of X if every X is a (kind of) Y
Y is a hyponym of X if every Y is a (kind of) X
Y is a coordinate term of X if X and Y share a hypernym
Y is s a holonym of X if X is a part of Y
Y is an eronym of X if Y is a part of X

Table 1: WordNet's Relations for Nouns

2.4. ConceptNet

ConceptNet is a commonsense knowledgebase, composed mainly from the Open Mind Project, written and put together by Hugo Liu and Push Singh (Media Laboratory Massachusetts Institute of Technology).ConceptNet is written in Python but its commonsense knowledgebase is stored in text files. It contains 1.6 million edges connecting more than 300 000 nodes at present. It contains lots of things computers should know about the world, especially when understanding text written by people. Until now, ConceptNet contains nearly one million of assertions represented as triplets like <concpet1, relation, concept2>, to define the concrete semantic relations between two specific concepts. There are some tools in text mining application area that use conceptnet for understanding the meaning of text and also for classifying the text.
According to Liu & Singh, 1984, ConceptNet, given a story

describing a series of everyday events, can infer where these events will likely take place; the mood of the story; and the possible next events. Furthermore they claim that, given a natural language search query, ConceptNet has the potential to determine which meaning is most likely. They also say that when presented with a novel concept appearing in a story, ConceptNet can determine which known concepts most closely resemble or approximate the novel concept. For example, for getting the similar and related words of the word "apple", easily we can search in ConceptNet and the result is shown in the figure 2.



```
[apple]
**OUT:********
  ==ConceptuallyRelatedTo==> person (141, 3)
  ==ThematicKLine==> grape (85, 0)
  ==CapableOfReceivingAction==> eat (5, 55)
  ==ThematicKLine==> orange (58, 0)
  ==ThematicKLine==> glass (55, 1)
  ==ThematicKLine==> strawberry (50, 0)
  ==IsA==> fruit (31, 6)
  ==ThematicKLine==> pineapple (34, 0)
  ==ConceptuallyRelatedTo==> glass (31, 1)
  ==ThematicKLine==> banana (32, 0)
  ==ThematicKLine==> juice (31, 0)
  ==ThematicKLine==> kiwi (31, 0)
  ==ThematicKLine==> leave (30, 0)
  ==ConceptuallyRelatedTo==> fruit (24, 3)
  ==ThematicKLine==> fruit (22, 2)
  ==ConceptuallyRelatedTo==> tree (23, 0)
  ==ConceptuallyRelatedTo==> apple (18, 4)
  ==ThematicKLine==> stem (20, 0)
  ==ConceptuallyRelatedTo==> apple juice (18, 1)
  ==ConceptuallyRelatedTo==> basket (18, 0)
  ==ConceptuallyRelatedTo==> background (18, 0)
  ==CapableOf==> grow on tree (16, 0)
  ==ConceptuallyRelatedTo==> day (16, 0)
  ==ThematicKLine==> ice cream (15, 0)
  ==ConceptuallyRelatedTo==> doctor (15, 0)
  ==ConceptuallyRelatedTo==> color (8, 6)
  ==ThematicKLine==> table (13, 0)
  ==ThematicKLine==> apple juice (13, 0)
```

Figure 2: Similarity with ConceptNet

2.5. ConceptNet Vs WordNet and Cyc

Unlike other projects like WordNet or Cyc, ConceptNet is based more on Context (Junpeng Chen, Juan Liu, 2011). ConceptNet makes it possible, within the limits of its knowledgebase, to allow a computer to understand new concepts or even unknown concepts by using conceptual correlations called Knowledge-Lines (K-Lines: a term introduced by Minsky, cf. The Society of Mind (1987)). K-Lines may be thought of a list of previous knowledge about a subject or task. ConceptNet actually puts these K-Lines together using it's twenty relationship types that fall into eight categories (including K-Lines)to form a network of data to simulate conceptual reasoning. What really makes all this possible is ConceptNet's Relational Ontology (the eight categories and twenty relationship types).

ConceptNet's power of linking subjects together is attributed to twenty relationship types defined by its Relational Ontology.

A partial snapshot of actual knowledge in ConceptNet is given in Fig 3.

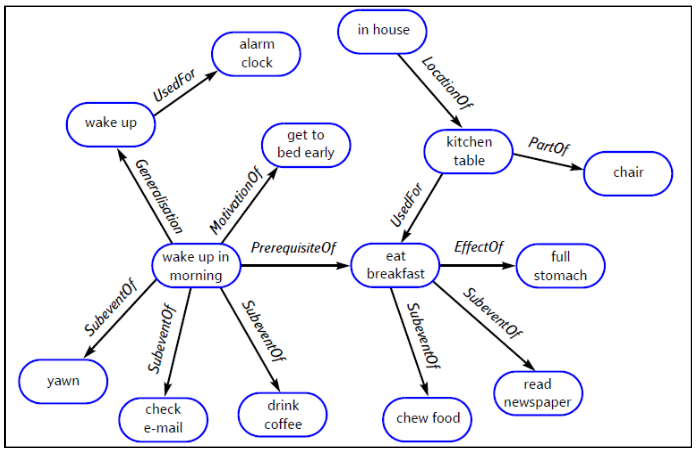


Figure 3: partial snapshot of ConceptNet actual knowledge

3. PROPOSED APPROACH

In this section we describe our approach for matching ontologies based on their lexical and semantic similarities. The proposed methodology contains three phases to find corresponding data between two given source and target ontologies. First and second steps, the given ontologies are lexically and semantically matched. In the semantic matcher phase, we use the ConceptNet to recognize the Concepts in each ontologies and also analysis the concepts, i.e.: classes, data properties, object properties and instances. For classifying the entities we use four matrices and update the similarity matrices when comparing and checking the given ontologies.

3.1. Pre-processing (Lexical matching)

Lexical similarity matching approaches are string-based methods for identifying similar entities in given ontologies. Here, we separately find the lexical similarity among entities (named classes, object properties, data properties and instances) of the source and target ontologies. This phase will produce four lexical similarity matrices as its output.

So Pre-processing phase, extracts each concept from source ontology, and then perform string matching and stemming and finally classify each elements of given ontologies into the corresponding matrices.

```
Function LexicalMatching(String A,String B)
{
        String s1,s2;
        s1=Stemming(A);
        s2=Stemming(B);
        if(StringCompare(s1,s2))
        return true;
        else
        return false;
}
```

lexical matching of items in ontologies

```
for(C ∈O_1 ,,, O_n)
{
if (type(c)="class") then
        add c to groupscls
else if (type(c)="Object property") then
        add c to groupsObjprops
else if (type(c)="Data property") then
        add c to groupsDTjprops
else if (type(c)="Instance") then
        add c to groupsInstance
}
```

classify the elements of ontologies

First the two ontologies will be given as an input in the lexical matching and by using the string matching will find the specific classes, properties and instances that are similar to each other.

We introduce a new similarity distance measure to determine the lexical similarity of input ontologies. Assume that we want to calculate lexical similarity between two strings s and t. These strings can be a single word or a text containing several statements. At first, we tokenize each string using delimiters and then convert it to a bag of words. Any non-alphabetical character in the given strings will be considered as a delimiter (J.B. Lovins, 1968). For example, if string s contains two non-alphabetical characters then we consider these two as delimiters and remove them from string s. As a result, s will be tokenized into three words. After converting each of the strings s and t to a bag of words, every word that is common to the two bags will be removed from both of them. After that, if there is nothing left in the first and second bags, the distance between the two input strings will be zero. Otherwise, all characters left in each bag will be concatenated and the Levenshtein similarity distance (Ismail Akbari and Mohammad Fathian, 2010) is calculated between the two resulting words. After calculating the distance between strings s and t, their similarity will be obtained from the following equation:

$$Lexical\_similarity(s,t) = 1 - \left\{ \frac{distance(s,t)}{max\_len(s,t)} \right\}$$

Where distance is the distance between strings s and t as described above, and max_len is the maximum of lengths of strings s and t.

3.1. Semantic Matching by using ConceptNet

In this phase the recognition of each concept in each ontology is doing by ConceptNet. Since in the pre-processing phase, each element of ontologies will be classify into the corresponding matrices, so here, the corresponding matrices in each ontologies match with each other and the comparison values put in the matrices.

```
Function SemanticAnalysis(String c1,String c2){
String s[];
bool m;
s=ConceptNet(c1);
m=false;
foreach string c1 in s
{
m=LexicalAnalysis(w,c2);
if(m)
    break;
}return m;}
```

Semantic analysis by using ConceptNet

We calculate the similarity for each matrices and then finding the items which are similar to each other. For example, in class matrix, we compare the class name of each ontologies with each other and then finding the corresponding classes. We consider two nodes in an ontology as neighbors if they are related to each other through one of 'is-a' (subclass or superclass), 'equivalent to' or 'disjoint with' relations or through an object property relation. For example, if class A is a subclass of class A', then A and A' are neighbors. For another example, given an object property p, its domain and range nodes will be neighbors. Based on this assumption, we calculate a neighbor matrix for any given ontology. Each element of the neighbor matrix is either 1 or 0 which shows that its row and column nodes are or are not neighbors, respectively. We do same job for other matrices and compare them with each other.

Every property of the class is compared with the other. To perform the comparison again lexical and semantic analysis is used. The number of similarities found is calculated and divided by the total number of properties. This value is useful for determining how much a class is similar to the other class.

```
Float PropertySimilarity (Class c1,Class c2,String p3[])
{
String p1[],p2[],p3[];
int a,c=0;float p;
bool m,n;
p1=c1.properties;
p2=c2.properties;
a=max(p1.length,p2.length);
foreach String q in p1
{
foreach String e in p2
{
m=LexicalAnalysis(q,e);
if(!m)
n=SemanticAnalysis(q,e);
if(m || n)
{
p3[c]=q;
c++;
}
}
}
p=c/a;
return p; }
```

measure the similarity for properties

The advantage of ConceptNet in the system is that, it does not necessary to do structural comparison, since ConceptNet is more accurate and consists of analogy part to compare the concepts in the each ontology.

4. EXPIEMENTS AND RESULTS

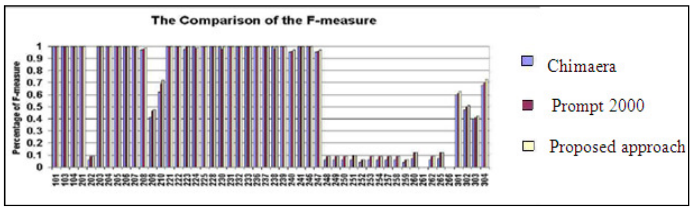
We evaluated the performance of the proposed algorithm using the quality factor. For the quality of matching results, we compared our results with three algorithms presented in the Ontology Alignment Evaluation Initiative 2011 (OAEI-11) and also with the Chimaera and Prompt. The OAEI is an annual campaign for ontology matching systems to identify their strengths and weaknesses. It is a coordinated international initiative that organizes the evaluation of an increasing number of ontology matching systems. Its main goal is to compare systems and algorithms on the same basis and to allow anyone to draw conclusions about the best matching strategies. The evaluation organizers provide a systematic benchmark test suite with pairs of ontologies to align as well as expected (human-based) results. These ontologies are described in OWL-DL and serialized in the RDF/XML format. The expected alignments are provided in a standard format expressed in RDF/XML and described in. I have developed a tool based on the proposed algorithm and applied it to the OAEI-11 benchmark test suite. We used standard information retrieval metrics to assess the results of my tests with (OAEI-2011):

$$precision = \frac{\#correct\_found\_alignments}{\#found\_alignments}$$

$$Recall = \frac{\#correct\_found\_alignments}{\#existing\_alignments}$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

We used the Jena package (http://jena.sourceforge.net) in our prototype tool to parse ontologies of the OAEI-11 benchmark test suite and also create a webservice in .Net for creating the ConceptNet (http://web.media.mit.edu/~hugo/conceptnet/) application and then invoke the application in java.



5. COMPARISON BETWEEN EXISTING METHODS WITH PROPOSED APPROACH

The expected performance of the proposed approach for ontology mapping is compared with the existing algorithm such as Chimaera, and PROMPT. The strengths of the proposed algorithm are it is fully automatic, there is no user interaction, it uses four different strategies and even if one of the strategies fails to acknowledge the match the other strategy will accomplish the task, and also by using to ConceptNet the proposed approaches is more accurate.

No	Parameters	Chimaera	Prompt 2000	Proposed approach
1	String matching	No	Yes	Yes
2	Semantic matching	Yes	Yes	Yes
3	Structural matching	Yes	No	No
4	Automation	Semi	Semi	Fully
5	Additional Resource	NA	NA	ConceptNet
6	Matching is based on	---	High Value	Classes, properties, instances

Table 2: Comparison between existing approaches with the

proposed approach

6. CONCLUSION AND FUTURE WORK

In this paper we presented an ontology matching algorithm that finds correspondences among entities of given ontologies based on their lexical and semantic information. This algorithm works at two phases: lexical and semantic. For determining lexical similarity among entities, we introduced a new similarity measure which converts each entity's available lexical information, such as its label or description, to a bag of words which are then used for finding their similarities. In the first phase we obtain four lexical similarity matrices by comparing named classes, object properties, data properties and instances of the two ontologies. In the second phase, to semantically compare ontologies, we use the ConceptNet for each node in the source and target ontologies and then compare them based on their grids. We have performed our algorithm on the benchmark test suite of the OAEI-2011 campaign and have had promising results. Also we compared our algorithm to some of the systems which participated in the OAEI-2011 contest and. In future we plan use Swoogle (http://swoogle.umbc.edu/), for selecting ontologies available online and apply the proposed approach to them.

ACKNOWLEDGMENT

This study was supported by Development of Real-Time Manufacturing Process & Integrated Equipment Management Technology for the Order-Based Industries (project id: 10035509) project sponsored by MKE (Ministry of Knowledge Economy) in Korea.

REFERENCES

Aldo Gangemi, Geri Steve, Fabrizio Giacomelli,(2007), ONIONS: An Ontological Methodology for Taxonomic Knowledge Integration", *Reparto Informatica Medica, Istituto Tecnologie Biomediche, CNR, Roma, Italy*

B. Ganter, R. Wille. (1997*),* Formal Concept Analysis, *Mathematical Foundations,Springer-Verlag New York, Inc., Secaucus, NJ, USA.*

C.D. Price. (1990), Another stemmer, *ACM Forum 24 (3)* 56–61.

Choi, N., Song, I.Y., Han, H(2006), A survey on ontology mapping. *SIGMOD Record*. 35(3), 34–41

D. McGuinness, R. Fikes, J. Rice, and S. Wilder(2000), The Chimaera Ontology Environment, *In Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*

G. Birkoff. (1976), Lattice Theory, *American Mathematical Society, Providence.*

H Liu and P Singh(2004), ConceptNet a practical commonsense reasoning tool-kit,*MIT press*

Ismail Akbari and Mohammad Fathian (2010),A novel algorithm for ontology matching, *Journal of Information Science*

Jérôme Euzenat, Pavel Shvaiko. (2007), Ontology matching,*Springer*

J.B. Lovins.(1968), Development of a stemming algorithm, *Mechanical Translation and Computational Linguistics* 11, 22–31.

J.L. Dawson(1974), *Suffix* removal and word connation, *ALLC Bulletin 2 (3)* ,33–46.

Junpeng Chen, Juan Liu(2011),Combining ConceptNet and WordNet for Word Sense Disambiguation, *Proceedings of the 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand*

Mehdi Keshavarz, Yong-Han Lee(2011),Ontology-based knowledge management in multi agent system, *APIEMS,Bejing,China*

Mikalai Yatskevich and Fausto Giunchiglia(2006), Element level semantic matching using WordNet, *Dept. of Information and Communication Technology, University of Trento*

Muhammad Abulaish. (2009),An Ontology Enhancement Framework to Accommodate Imprecise Concepts and Relations, *Journal of Emerging technologies in web intelligence*, VOL .1,NO .1

Natalya Fridman Noy and Mark A. Musen(2000), PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment, *American Association for Artificial Intelligence*

Natalya F. Noy and Mark A. Musen(2006),Anchor-PROMPT: Using Non-Local Context for Semantic Matching, *Stanford Medical Informatics, Stanford University, Stanford, CA 94305-5479*

Pan, R., Ding, Z., Yu, Y., Peng, Y(2005),A bayesian network approach to ontology mapping. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 563–577. *Springer, Heidelberg*

Rung-Ching Chen, Cho-Tscan Bau, Chun-Ju Yeh.(2011), Merging domain ontologies based on the WordNet system and Fuzzy Formal Concept Analysis techniques, *Applied Soft Computing journal*

R. Wille(1982), Restructuring lattice theory: an approach based on hierarchies of concepts, in: I. Rival (Ed.), *Symposium on Ordered Sets, Reidel.*

Shvaiko, P., Euzenat, J.(2005): A survey of schema-based matching approaches. *Journal of Data Semantics* 4, 146–171

Yves R. Jean-Mary, E. Patrick Shironoshita, Mansur R. Kabuka(2009), Ontology matching with semantic verification, *WEBSEM-158*; No. of Pages17

AUTHOR BIOGRAPHIES

Mehdi Keshavarz received the M.S. degree in Information systems from the Osmania University, India, in 2009, and he is currently working toward the PhD degree in Dongguk University. His research is about Multi agent systems and Ontology mapping in industrial domain. His email address is <keshavarzm2005@gmail.com>

Yong-Han Lee is an associate professor in Industrial and Systems Engineering department at Dongguk University-Seoul since 2003. His current research interests include RFID applications in manufacturing processes, business process management, service oriented development of applications, and enterprise modeling and simulation. He received his BS, MS, and Ph.D. degree from Seoul National University, KAIST, and Penn State, respectively. His email address is <yonghan@dgu.edu>