Using Unstructured Documents as Background Knowledge for Ontology Matching

Chanakarn Kingkaew

***Abstract*—O**ntology Matching is the process of finding correspondences between two given ontologies. Many traditional matching techniques have been proposed and the majority of approaches rely on lexical and structural matching techniques which need a sufficient lexical overlap and a rich structure. There is a new approach using background knowledge for ontology matching. This method provides a way to find matches missed by the other approaches. There are various approaches using background knowledge, but most of them require that the background knowledge is explicitly given as input to the matching process. However, the background knowledge has to be provided as a formal ontology which is often not available. Our work extends this approach by considering unstructured documents as background knowledge.

We evaluate our matcher with the sample data set from the conference track of the Ontology Alignment Evaluation Initiative (OAEI) campaign. We use precision, recall and f-measure as our metric for evaluating the results. The evaluation shows that our matcher overcomes the traditional matching techniques in every case.

***Keywords*—** Semantic Web, Ontology Matching, Natural Language Processing, Background Knowledge.

I. Introduction

SCHEMA and Ontology matching is widely used in data management in various application domains, such as semantic web, bio-medical [9], data mining, e-commerce, query mediation, etc. The aim is to identifying semantic correspondences between metadata structures or models such as database schemas, XML message formats, and ontologies. Solving such match problems are of key importance to service interoperability and data integration.

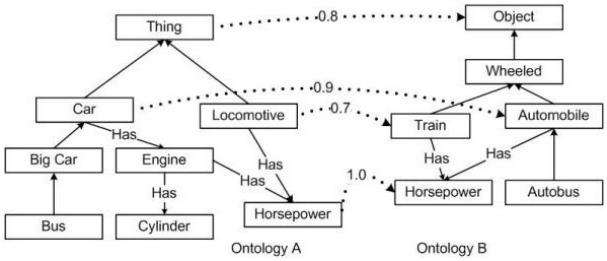

Fig. 1 The example of ontology A and B to demonstrate how each component matches together.

Chanakarn Kingkaew is a master degree candidate of The Sirindhorn International Thai-German Graduate School of Engineering (TGGS), King Mongkut's University of Technology North Bangkok (KMUTNB) under the supervision of Dr. Christoph Quix and Prof. Dr. Matthias Jarke of the Information Systems & Databases group (Informatik 5) RWTH Aachen University, Aachen, Germany. E-mail: kingkaew@dbis.rwth-aachen.de phone: +66891089811

A Match operation, as defined in [4], takes two schemas as input and produces a mapping between elements of the two schemas that correspond semantically to each other like Fig 1. To reduce the manual effort required, many techniques and prototypes have been developed to semi-automatically solve the match problem such as FALCON-AO [6] or GeRoMeSuite [7].

II. Problem Specification

The problem of ontology matching has a central role in the development of knowledge based systems. New technologies such as Semantic Web make it easier to use ontologies in information systems. These trends have driven the development of new ontologies, and that resulted in an increasing amount of ontologies becoming available in the recent years. An essential characteristic of ontologies is reusability; to use an existing ontology in a newly developed system one needs to integrate it into the system.

It has been stated that the lack of background, most often domain specific knowledge, is one of the key problems of matching systems these days [13].

Various approaches for schema and ontology matching have been developed [12], and they mainly focus on two aspects: lexically matching the elements of the ontologies, and using the structure of the ontologies. The first uses string-based and linguistic methods to detect relatedness between elements based on string similarity of their labels, and the second uses the relations within the ontologies to detect similarities. Hence, the majority of approaches crucially rely on two assumptions:

– Sufficient lexical overlap exists between the source and target ontology

– Source and target ontology have sufficient structure

A big limitation in both of them appears if the elements in the ontologies are related but neither have neither lexical similarity nor the structure of the ontologies provides evidence of relatedness. Motivated from this issue, we focused on using background knowledge in form of ontology when matching ontologies [11]. We followed the fact that a background ontology which comprehensively describes the domain will provide a way to find matches missed by other approaches as discussed in [2].

However, especially in the field of biomedicine conceptual knowledge is scattered over various different, often disconnected ontologies. While some of them semantically overlap (such as two different anatomy ontologies), others complement each other rather by design (such as ontologies

for anatomical structures, cells, proteins, biological processes, drugs and diseases) [3, 14]. Also, one important drawback is that using background knowledge assumes the presence of background ontology. Again, the work in [11] has the limitation that appropriate background ontology must be present in the Web, but Background knowledge is not always available as a formal ontology.

We propose a different approach to detect many different types of semantic relations not within the given formal ontologies themselves but in large text collection across many domains such as websites, online documents, and even a broad-coverage text corpus such as Wikipedia. However, relations are hidden in natural language processing (NLP) and an appropriate NLP system is required to access them.

III. Solution Approach

Our approach is similar to the approach presented in this section. The main different is that we find semantic information from unstructured data such as text from web. The proposed method in [2] consists of finding semantic matches by using a (richly structured) ontology that holds background knowledge about the domain. First, the source and target vocabulary are each matched with the background knowledge ontology producing so-called anchoring matches. Anchoring matches connect a source or target concept to one or more concepts in the background knowledge ontology, which we call anchors. Then, based on the relationships among the anchors entailed by the background knowledge, they induce in a second step how the concepts from the source are matched to the concepts in the target vocabulary, yielding the semantic match we are looking for. A concept can match to several anchors, in which case the collection of anchors is combined according to the semantics of the background knowledge as in fig. 2.

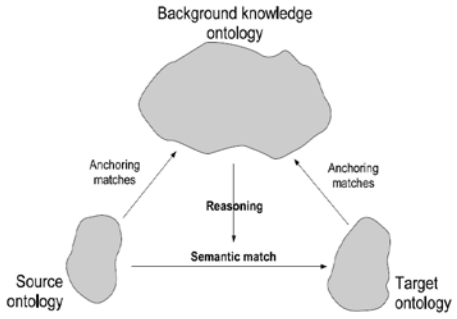

Fig. 2 Background Knowledge for Ontology Matching

The main objective in this work is to develop a matcher which will use unstructured documents as Background Knowledge. In fig. 3, we show an overview of the system which consist of 3 main components. First, Information Retrieval component has a main function to query unstructured documents based on ontologies to use as Background Knowledge. Then, Natural Language Processing component has a main function to process unstructured documents, extract relationship and represents them as Similarity Graph. Lastly, we match source and target ontology with Background Knowledge using Similarity Graph and the result is the matching morphism.

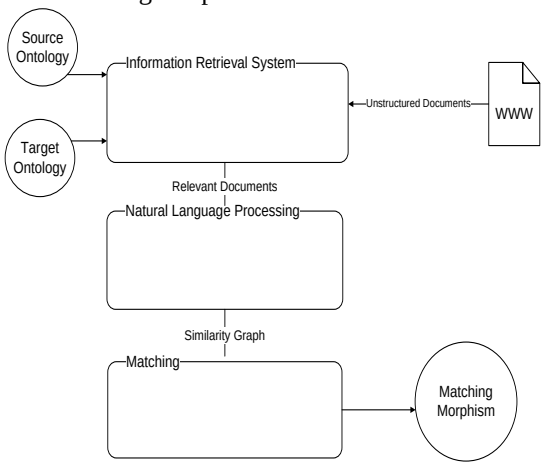

Fig. 3 Overview of Solution Approach

We describe our approach of Matcher using unstructured Documents as Background Knowledge. We will demonstrate step by step in given sample scenario. Suppose we want to match ontology A and Ontology B as in fig. 1.

A. *Information Retrieval System*

The Information Retrieval System has function to query sets of sentences based on keyword given by source and target ontologies. It first extracts source and target elements into the bag of words. We remove 'Stop words' and calculate Term-Frequency (TF), Term-Frequency Inverse Document Frequency (TF-IDF) to find the best document represent all elements in both source and target ontologies. Also, Random-K method to selects K words randomly to query. Then, Search API queries the World Wide Web (WWW) from generated keywords and then select limit number of documents and store into the document database.

Because we cannot change the result ranking in the Search Provider, search length is limited, and quota to query is limited, we cannot compare the similarity of retrieve document with respect to source and target ontologies we will select the most relevant documents using Apache Lucene by using our own similarity score.

We parse documents into Apache Lucene to calculate similarity score and query for the most relevant documents with respect to source and target ontologies. We need documents which can represent both source and target ontologies then we calculate similarity using Lucene Similarity scoring method. We propose the equation that give the similarity value that relevant to both source and target ontologies as equation below.

Similarity of source ontology s and target ontology t respect to document d is described as [QRK11]:

$$Sim(s,t,d)=\alpha(Sim(s,d)+Sim(t,d))-\beta|(Sim(s,d)-Sim(t,d))|$$

where $\alpha > \beta$

TABLE I
THE EXAMPLE OF THE SELECTION OF DOCUMENTS TO USE AS BACKGROUND KNOWLEDGE

Document d	Lucene Similarity Score *Sim(s,d)*	Lucene Similarity Score *Sim(t,d)*	*Similarity Sim (s,t,d)* Where $\alpha = 1.0$ and $\beta = 0.9$
1	0.19	0.79	0.44
2	0.5	0.49	0.891
3	0.97	0.37	0.8

We can see from table I that the equation give us the similarity value between both source and target ontology. Document 3 has highest Lucene similarity score to source ontology but less in target ontology but Document 2 has the best score since the equation give us the highest Similarity value in this case. We select documents where similarity is greater than 0.5.

B. *Natural Language Processing*

The retrieved texts from the Information Retrieval System will use in this step. First, the Sentence Detector will break a document into a sentence. Given input is a document and output is a set of sentences.

Then, each word in a sentence will be split into a token using the Tokenizer. Given input is a sentence and output are tokens.

The Part-of-Speech Tagger for matching has the function to tag each word in a sentence with correspond part-of-speech. Brown Corpus [1] and Penn TreeBank tag set [10] describe the details for each part-of-speech tag. Given tokens as input and output are tokens which already tag with correspondence part-of-speech as in Table II.

TABLE II
THE PART-OF-SPEECH TAGGER

Input	Output
[A] [locomotive] [is] [a] [railway] [vehicle] [that] [provides] [the] [motive] [power] [for][a] [train][.]	A_DT locomotive_NN is_VBZ a_DT railway_NN vehicle_NN that_WDT provides_VBZ the_DT motive_JJ power_NN for_IN a_DT train_NN .

Then, the Internal Similarity Graph Generator will generate similarity graphs that represent the semantic relationship using the similarity degree between source keywords and target keywords. We demonstrate by focus on keyword "Locomotive". There are 2 solutions proposed.

1) POS Method

The example explains how Similarity measure is calculated as in Table II. Internal Similarity Graph Generator will generate similarity graph as fig.4. By using POS tagger we will find the correspondence pair of noun. We must first calculate the occurrence of each pairs of noun as in Table IV.

2) Pair Frequency

In this solution, instead of calculate pairs of noun. We calculate the occurrence of 2 source and target keywords in a single sentence. For example, consider a sentence "a locomotive is wheeled vehicle consisting of a self-propelled engine that is used to draw trains along railway tracks". We can see that Locomotive and Wheeled occur together and also locomotive and engine in this sentence. Suppose we get the similarity graph as fig.4.

TABLE III
THE EXAMPLE SIMILARITY GRAPH CALCULATION USING PART-OF-SPEECH (POS) TAGGER TO FIND PAIRS OF NOUN IN A SENTENCE (POS METHOD)

Input as Part-of-Speech[1]	Output as pairs of nouns <x,y> where x is noun and y is noun.
A_DT locomotive_NN is_VBZ a_DT railway_NN vehicle_NN that_WDT provides_VBZ the_DT motive_JJ power_NN for_IN a_DT train_NN.	<locomotive,railway> <locomotive,vehicle> <locomotive,train>
A locomotive has no payload capacity of its own, and its sole purpose is to move the train along the tracks.	<locomotive,train> <locomotive,track>
Traditionally, locomotives pull trains from the front.	<locomotives,train>
The locomotive only ran three trips before it was abandoned.	<locomotive,trip>
The first successful locomotives were built by Cornish inventor Richard Trevithick.	<locomotives,inventor>
A locomotive would be like a train or something of that nature.	<locomotive, train> <locomotive,nature>

TABLE IV
THE PART-OF-SPEECH TAGGER (POS METHOD)

Pair of "Locomotive"	Occurrence together with "Locomotive"	Similarity Degree = (Occurrences/Number of pairs)
<Locomotive,Train>	6	6/15 = 0.4
<Locomotive,Railway>	2	2/15 = 0.13
<Locomotive,Vehicle>	2	2/15 = 0.13
<Locomotive,Track>	2	2/15 = 0.13
<Locomotive,Car>	1	1/15 = 0.06
<Locomotive,Engine>	1	1/15 = 0.06
<Locomotive,inventor>	1	1/15 = 0.06
Total	15	1

C. *Matching*

Now we need to aggregate the External Similarity Graph with the Internal Similarity graph. There is case that keywords from source or target ontologies might not directly match with Internal Similarity Graph. We need direct lexical matching method such as JaroWinkler measure to match the keyword with the element of the Internal Similarity Graph as in fig. 4

[1] We omit Part-of-Speech tag from the first sentence so that it can read easily.

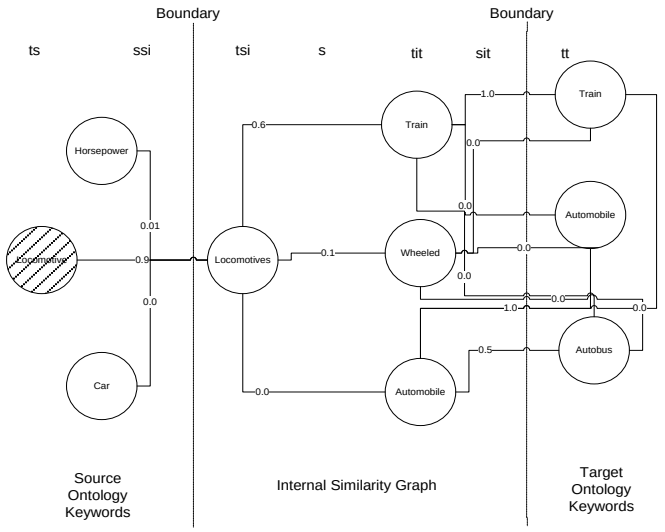

Fig. 4 External Similarity Graph with boundary to Internal Similarity Graph

A matching is 8-tuple relationships M = <id, t_s, s_{si},t_{si},s,t_{it},s_{it},t_t> where:

- id is a matching identification
- t_s is a homogenous ontology term of source ontology
- s_{si} is an external similarity degree from t_s to t_{si} where $s_{si} \in [0,1]$ calculate using Direct String Matching Method
- t_{si} is a homogenous term of an internal term which connected to t_s
- s is an internal similarity degree where $s \in [0,1]$
- t_{it} is a homogenous term of an internal term which connected to t_t
- s_{it} is an external similarity degree from t_{it} to t_t where $s_{it} \in [0,1]$ calculate using Direct String Matching Method
- t_t is a homogenous ontology term of target ontology

We match <t_s,t_t> if there is a relationship that satisfy these condition:

1) s_{si}, s_{it} *> External Bound Threshold (et) by default 0.5*
2) s > Internal Bound (it) by default 0.5

Then we will match <t_s,t_t> and use Internal Bound s as a similarity degree.

We implemented proposed matching approach in GeRoMe (A Generic Role Based Metamodel for Model Management) [7,8] framework and integrated our approach with it. The benefit of GeRoMeSuite is that it is rich of matching configuration from lexical matching to structural matching. Schemas that we are going to match can be in any modeling language such as XML, OWL and etc. Therefore, we need to convert them into a generic modeling language (GeRoMe metamodel) so that it can be compared.

IV. Evaluation

Evaluation of matching results is made on the basis measures that are precision and recall. In [5] precision and recall are originating from information retrieval. In context of the ontology matching, the terms true positives (tp), true negatives (tn), false positive (fn) and false negatives (fn) compare the predicted class with the actual class. To evaluate our system we need to compare our matching result (the predicted class) with the reference alignments (the actual class) that provide by the unbiased such as OAEI. Given a pair of ontologies, these algorithms compute a set of correspondences between entities of these ontologies, called alignment or morphism which is the result of matching.

Precision shows the correctness of match result or alignment (A) with the reference alignment (R) therefore precision is the ratio of the number of true positives over the total number of computed or predicted correspondences as (1).

$$Precision = \frac{|R \cap A|}{|A|} = \frac{tp}{tp+fp} \quad (1)$$

Recall shows the completeness of match result or alignment (A) with the reference alignment (R) therefore recall is the ratio of the number of true positives over the total number of actual correspondences as (2).

$$Recall = \frac{|R \cap A|}{|R|} = \frac{tp}{tp+fn} \quad (2)$$

As we discussed about precision and recall, in this section we introduce to F-measure which is the harmonic mean of precision and recall. The equation of F-measure is as (3):

$$F(\beta) = \frac{(1+\beta^2).(precision.recall)}{(\beta^2.precision+recall)} \quad (3)$$

The popular β value is equal to 1 because recall and precision are evenly weighted.

We will compare the result of matching with direct and structural matching approach such as OAEI2008, OAEI2009. The sample datasets that we use to test are from OAEI or Ontology Alignment Evaluation Initiative campaign. OAEI organizes evaluation campaigns aiming at evaluating ontology matching technologies. The sample dataset that we are going to use is from OAEI of year 2010 and the sample dataset called conference track which is a collection of ontologies is dealing with conference organization.

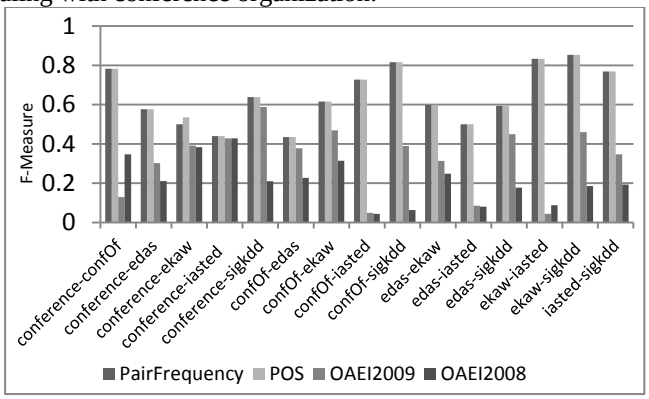

Fig. 5 External Similarity Graph with boundary to Internal Similarity Graph

From fig. 5, a comparison between a different matching configuration show us that our matching configuration 'POS'

and 'PairFrequency' achieved higher f-measure in all cases.

TABLE V
TIME USE IN MATCHING

Time Use in Matching	Seconds
POS	491.376
PairFrequency	82.35
OAEI 2009	0.333
OAEI2008	0.415

However, Table V shows us that pos method consumes more time than pairFrequency method which has the same result because we have to parse token in to part-of-speech tagger to analyze noun tag in addition from pairFrequency method. The speed is depends on the number of documents while OAEI2009 and OAEI2008 are not depends on the number of documents.

V. CONCLUSION

In this work we have developed a matcher which matches source and target ontologies by using unstructured documents as background knowledge. The main components of this matcher are an information retrieval system which retrieves unstructured documents from the WWW; then we process them using Natural Language processing phase. Finally we construct the similarity graph that represents those documents and use that graph as background knowledge for matching. The result that we achieved is a great success in terms of the quality, completeness and correctness. Surprisingly, POS and PairFrequency methods are able to achieve nearly the same f-measures.

However, the time performance of the matching process is a major consideration since it takes too much time to process documents.

There are several possibilities for future work. A complex Sentence is a sentence that gives us more than one meaning such as a sentence "A parrot is a kind of bird but it is not a mammal" in this case we can extract at least 2 semantic relationships such as <A parrot, is a, kind of birds> and < A parrot, is not, a mammal> so we can see that we have subsumption relationship on <A parrot, is a, kind of birds> and disjoint relationship < A parrot, is not, a mammal>. We can extend our work from POS tagging method by considering the verb which is in between two nouns to detect the semantic relationship between two concepts as indicated.

Furthermore, the evaluation has shown that the time consumption in the NLP process is a big concern in this research because it takes most of the time in the matching process.

ACKNOWLEDGMENT

I am sincerely and heartily grateful to my advisor, Dr. Christoph Quix, for the support and guidance he showed me throughout this work. I am sure it would have not been possible without his help. Besides I would like to thank to my family and my friends boosted me morally and provided me great information resources. I also would like to express my gratitude to Prof. Dr. Matthias Jarke who has me the opportunity to conduct this work in i5 laboratory in RWTH Aachen.

REFERENCES

[1] [ADH*00] Atwell, E S; Demetriou, G; Hughes, J; Schriffin, A; Souter, D C; Wilcock, S P:A comparative evaluation of modern English corpus grammatical annotation schemes. International Computer Archive of Modern and Medieval English Journal: Computers in English Linguistics, vol. 24, pp.7-23. 2000.

[2] [AKKH06] Z. Aleksovski, M. Klein, W. ten Kate, F. van Harmelen: Matching Unstructured Vocabularies using a Background Ontology In Proceedings of EKAW, 2006.

[3] [Beis10] E. Beisswanger: Exploiting Relation Extraction for Ontology Alignment. Doctoral Consortium ISWC 2010.

[4] [BLP00] P. A. Bernstein and A.Y. Levy and R. A. Pottinger: A vision for management of complex models. SIGMOD Record, Volume 29, page 2000, 2000.

[5] [Euze07] J. Euzenat. Semantic Precision and Recall for Ontology Alignment Evaluation. IJCAI 2007: 348-353. 2007.

[6] [JHCQ05] N. Jian, W. Hu, G. Cheng, and Y. Qu. Falcon-AO: Aligning ontologies with falcon. In Integrating Ontologies Workshop Proceedings, page 85. 2005.

[7] [KQCJ07] D. Kensche, C. Quix, M.A. Chatti, M. Jarke: GeRoMe: A Generic Role Based Metamodel for Model Management Journal on Data Semantics, Vol. VIII, LNCS 4380, 82-117, Springer-Verlag, 2007.

[8] [KQLL07] D. Kensche, C. Quix, Xiang Li and Yong Li: GeRoMeSuite: A System for Holistic Generic Model Management. VLDB, page 1322-1325, 2007.

[9] [MBB06] F. Mougin, A. Burgun, O. Bodenreider: Mapping data elements to terminological resources for integrating biomedical data sources BMC Bioinformatics, 2006.

[10] [MSM93] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz: Building a Large Annotated Corpus of English: The Penn Treebank, in Computational Linguistics, Volume 19, Number 2 (June 1993), 313-330 (Special Issue on Using Large Corpora). 1993.

[11] [QRK11] C. Quix, P. Roy, D. Kensche: Automatic Selection of Background Knowledge for Ontology Matching. 3rd International Workshop on Semantic Web Information Management (SWIM 2011, in conjunction with ACM SIGMOD 2011), 2011.

[12] [RaBe01] E. Rahm, P. A. Bernstein: A survey of approaches to automatic schema matching, The VLDB Journal 10: page 334–350, 2001.

[13] [ShEu08] P. Shvaiko, J. Euzenat: Ten Challenges for Ontology Matching In Proceedings of ODBASE, 2008.

[14] [SAR*07] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L. Goldberg, K. Eilbeck, A. Ireland, C.J. Mungall, N. Leontis, P. Rocca-Serra, A. Ruttenberg, S.A. Sansone, R.H. Scheuermann, N. Shah, P.L. Whetzel, and S.E Lewis: The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration, Nature Biotechnology 25(11), page 1251-1255, 2007.