Eff2Match Results for OAEI 2010

Watson Wei Khong Chua[1] and Jung-Jae Kim[2]

[1] Nanyang Technological University, Singapore watsonchua@pmail.ntu.edu.sg
[2] Nanyang Technological University, Singapore jungjae.kim@ntu.edu.sg

Abstract. While the primary objective of an ontology alignment tool is to identify as many correct correspondences as possible, efficiency in terms of run-time needs to be achieved for practical usage. Not only does run-time efficiency enable scalability, it also facilitates information integration for time-critical applications using heterogeneous ontologies. In this paper, we present our ontology alignment approach known as *Eff2Match* which aligns a pair of ontologies with high accuracy and low runtime.

1 Presentation of the system

Ontologies are being widely used for semantic representation in applications from various domains such as biomedical informatics [3] and earth sciences [5]. They can be used to provide data with semantics, thus resolving the heterogeneity problem between information sources at the data level. However, the problem is only partially resolved because different ontology engineers model their ontologies differently. Therefore, the heterogeneity problem is escalated to the ontological level if information is to be shared between applications using different ontologies. As a result, ontology alignment tools that can achieve high accuracy are required. In this paper, we present *Eff2Match* (pronounced "Eff Squared match"), an **Eff**ective and **Eff**icient ontology matching tool which can match a pair of ontologies with good accuracy within a short amount of time. *Eff2Match* uses an effective and dynamic candidate reduction technique to avoid performing unnecessary comparisons, thereby achieving high efficiency.

1.1 State, purpose, general statement

In order to facilitate the sharing of information among applications using different ontologies, we have developed an automatic ontology alignment tool called *Eff2Match*. It is able to align both concepts and properties in different ontologies that are semantically equivalent (concepts are matched to concepts and properties are matched to properties). The current implementation does not match the instances in the ontologies.

1.2 Specific techniques used

Eff2Match takes as input the URI of a pair of ontologies to be aligned and matches entities (concepts or properties) in the source ontology to those in the target ontology. The alignment process consists of four stages: 1) Anchor Generation, 2) Candidates Generation, 3) Anchor Expansion and 4) Iterative Score Boosting as shown in Fig. 1 .

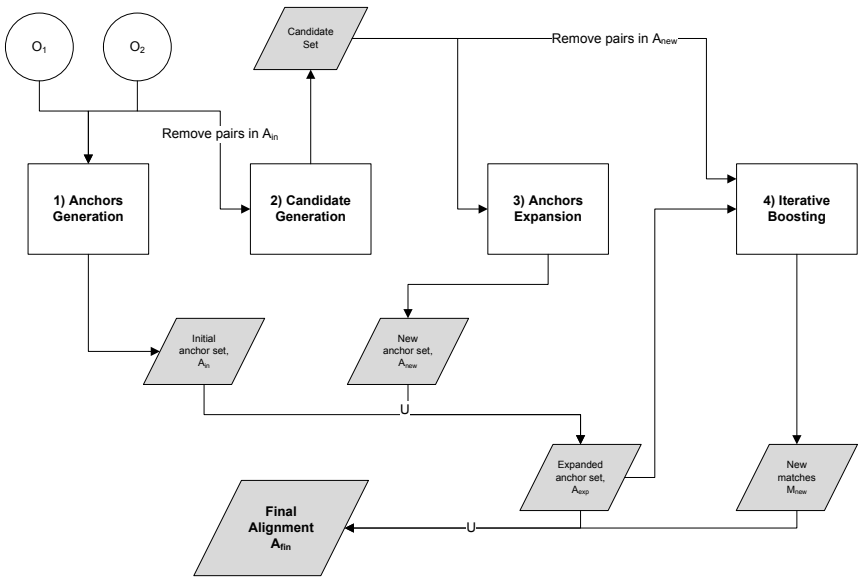

Fig. 1. Eff2Match Algorithm Flow

Anchor Generation In the Anchor Generation stage, matching entities are identified using an exact string matching technique. Local names and labels of entities $e2_j$ in the target ontology are first preprocessed through camel case conversion, case-normalization and removal of delimiters. A hash-table is then used to map the preprocessed local names and labels to their corresponding entities. After that, we preprocess the local name and label for each entity $e1_i$ in the source ontology and look them up in the hash-table. If either a matching local name or label can be found in the hash-table, we consider the corresponding entity $e2_j$ in the target ontology to be equivalent to the source entity. This method is significantly faster than a pairwise comparison of local names and labels as it takes only $O(n_1) + O(n_2)$ time compared to the latter which requires $O(n_1 \times n_2)$ where n_1 and n_2 are the number of entities in the source ontology O_1 and target ontology O_2 respectively.

Candidate Generation In the Candidates Generation stage, we enumerate candidates for entities in the source ontology that has not been matched in the previous stage using a Vector Space Model (VSM) approach. For each concept, we generated three VSM vectors from the annotations (local name, label and comments) in the ancestors (Vec_a), descendants (Vec_d) and the concept (Vec_c) itself. For each property, the vectors generated consist of the annotations in the property (Vec_p) itself, the property's domain concepts (Vec_{do}) and range concepts(Vec_r). The VSM similarity (Sim_{VSM}) between two concepts $c1_i$ and $c2_j$ is an aggregation of the cosine similarity between the concept VSM vectors, ancestor VSM vectors and descendant VSM vectors using a weighted

average.

$$Sim_{VSM} = \frac{\alpha \times Vec_{c1i} \cdot Vec_{c2j} + \beta \times Vec_{a2i} \cdot Vec_{a2j} + \gamma \times Vec_{d2i} \cdot Vec_{d2j}}{\alpha + \beta + \gamma}$$

where α, β and γ are the weights given to the similarity between annotations of the concepts themselves, annotations of their ancestor concepts and annotations of their descendant concepts respectively. The similarity values for properties are calculated in a similar manner and two matrices, M_{con} and M_{prop} are used to store the similarity values for concepts and properties respectively. The VSM similarities are normalised to [0,1] by dividing each entry in M_{con} and M_{prop} by their largest value to get the normalised VSM similarity, $Sim_{VSM_N}(C1_i, C2_j)$. Candidates selection is then performed for each source entity by taking the top-K entities in the target ontology according to their VSM similarities.

Anchor Expansion In the anchor expansion stage, more equivalent pairs of entities are identified by comparing the source entities with their candidate entities using terminological methods. In *Eff2Match*, a term-removing algorithm (TRA) is used for efficiency purposes and the algorithm is illustrated in Fig. 2. First, the labels (local names) of a pair of entities are tokenised. The tokens are then stemmed and words that are stemmed to the same form are removed from both labels. If there are tokens remaining in both the labels, the next stage compares tokens from different labels pairwise using WordNet to determine if they are synonyms of each other. Synonymous tokens are then removed from the labels. If there are no tokens remaining in both the labels after any stage, the two entities are considered to be equivalent and added to the anchor set.

If there are remaining tokens in only one of the labels and not the other, we use a novel technique known as Informative Word Matching (IWM) to determine if the entities are matches. If concept $C1_i$ contains p more terms than $C2_j$ and these p terms occur in the labels of the ancestors of $C2_j$, we can consider $C1_i$ and $C2_j$ to have the same meaning. For example, if $C1_i$ has the label *Heart Endocardium* and $C2_j$ has the label *Endocardium* and the $C2_j$ is a sub-concept of *Heart Part*, we can determine that $C1_i$ and $C2_j$ are semantically equivalent from their labels since the word *Heart* in $C1_i$ is not informative.

Given a pair of entities e_{emp} and e_{rem} where e_{emp} is the entity without remaining tokens in its label and e_{rem} is the entity with p remaining tokens in its label, the following steps are performed to determine if the p remaining tokens are informative words:

1. Collect the labels of ancestors of e_{emp} up to r generations or when the root of the ontology is reached, whichever is earlier.
2. Tokenise and stem the collection of labels collected to get a set of stemmed ancestor tokens S_{sat}.
3. For each token t_i, $i \in [1..p]$ remaining in e_{rem}, stem t_i and check if it exists in S_{sat}. If it does, the word is not informative and it is removed it from e_{rem}.
4. Look up the definition of the original label of e_{emp} in WordNet, tokenise and stem the words in the definition before adding them to the set of definition words, S_{def}.

5. For each token t_i, $i \in [1..p]$ remaining in e_{rem}, stem t_i and check if it exists in S_{def}. If it does, the word is not informative and it is removed it from e_{rem}.

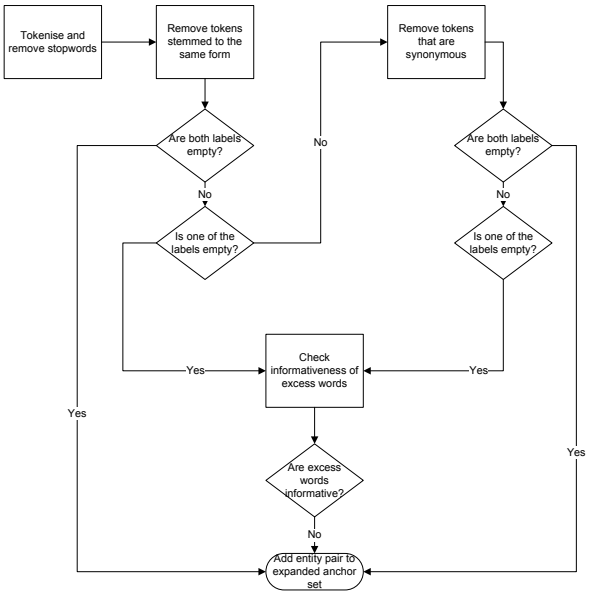

Fig. 2. Anchor Expansion Flow Chart

Iterative Boosting In the final stage of the matching process, an iterative boosting (IterBoost) process is used to identify more pairs of equivalent concepts using the expanded anchor set A_{exp}. In this stage, the algorithm attempts to match the source concepts that have not been matched with their candidates iteratively. In each iteration, the source concepts are ranked based on the sum of their ancestors and descendants with matches in A_{exp}. The top K source concepts are then selected and a formula is used to boost the score of their candidates based on the number of common ancestors and descendants that they share. Given a source concept $C1_i$ and a candidate concept $C2_j$, the structural overlap $SO(C1_i, C2_j)$ between them is calculated using:

$$SO(C1_i, C2_j) = \frac{|\xi(A(C1_i), A(C2_j))| + |\xi(D(C1_i), D(C2_j))|}{min(|A(C1_i)|, |A(C2_j)|) + min(|D(C1_i)|, |D(C2_j)|)}$$

where $A(C)$ is the set of ancestors of concept C and $D(C)$ is the set of descendants of concept C and the function $\xi(X, Y)$ enumerates the set of concepts in X which have equivalences in Y. The equivalences were determined from the comparison in the

previous stages. The similarity between $C1_i$ and $C2_j$ is then given by:

$$Sim_{boost}(C1_i, C2_j) = \begin{cases} Sim_{VSM_N}(C1_i, C2_j), & SO(C1_i, C2_j) > t_b \\ \sqrt{Sim_{VSM_N}(C1_i, C2_j)}, & otherwise. \end{cases}$$

The highest scoring candidate is then selected to be the matching concept and the confidence that the pair matches is given by $Sim_{boost}(C1_i, C2_j)$. If $Sim_{boost}(C1_i, C2_j)$ is greater a cut-off threshold t_c, $C1_i$ and $C2_j$ are inserted into A_{exp} as well as the final set of alignment and the process is repeated until all the source entities and their candidates have been visited. If $Sim_{boost}(C1_i, C2_j)$ is less than t_c, $C1_i$ and $C2_j$ are inserted into the set of final alignment but are not considered anchors for future iterations.

1.3 Adaptations made for the evaluation

No special adaptations were made for individual tracks and all alignment processes make use of the same set of parameters. The cut-off threshold for the correspondences was set at 0.7 for the best F-Measure. The only external resource that we used is WordNet. For Informative Word Matching, we set $p = 1$, meaning that we only perform IFM for entities with one remaining token after the Term-Removing Algorithm (TRA). For *IterBoost*, the cut-off threshold for boosting, t_b is set at 0.4 while the cut-off threshold for matching entities to be considered anchors, t_c is set at 0.5. Lastly, the weights α, β, and γ are set to 2, 1, 1, respectively. The matcher has also been implemented as a web service so that it can be evaluated on the SEALS platform.

1.4 Link to the system and parameters file

The Eff2Match system (jar file) and configuration files can be found at `http://www.cais.ntu.edu.sg/~chua0507/OAEI/Eff2MatchSystem.zip`

1.5 Link to the set of provided alignments (in align format)

The set of alignments produced by Eff2Match can be found at `http://www.cais.ntu.edu.sg/~chua0507/OAEI/Eff2MatchAlignments.zip`

2 Results

Eff2Match participated in the benchmark, anatomy and conference tracks of the OAEI 2010 competition and the results are presented in the following subsections:

2.1 Benchmark

The ontologies in the benchmark dataset can be categorised into 5 different categories according to the difficulty of matching as shown in Table 2.1. *Eff2Match* performs well on all the categories, achieving an F-Measure of more than 0.75 with the exception of the category $248 - 266$. Ontologies in this category have different linguistics and structural characteristics, which are core attributes which *Eff2Match* relies on for finding correspondences, thus explaining the poor results.

Ontologies	Precision	Recall	F-Measure
101-104	1.000	1.000	1.000
201-210	0.986	0.684	0.768
221-247	0.990	1.000	0.995
248-266	0.929	0.502	0.591
301-304	0.889	0.711	0.780

Table 1. Results for benchmark dataset

2.2 Anatomy

The anatomy dataset consists of two large real world ontologies, namely the Adult Mouse Anatomy with 2247 classes and the anatomy part of the NCI Thesaurus with 3304 classes. *Eff2Match*'s results for this track are shown in Table 2.2 and shows that *Eff2Match* can match large ontologies with high accuracy and short run-time.

Ontologies	Precision	Recall	F-Measure	Time Taken
mouse-human	0.955	0.781	0.859	2.5 mins

Table 2. Results for anatomy dataset

2.3 Conference

Lastly, Table 2.3 presents results for *Eff2Match* in the conference track for the 21 ontologies where reference alignments are available. As these ontologies are developed heterogeneously, discovering the correct correspondences between them is more difficult than for the other two tracks. *Eff2Match* was able to achieve F-Measures ranging from 0.4 to 0.759 for pairs of ontologies in this track.

3 General comments

3.1 Comments on the results

This is the first time that *Eff2Match* is participating in the OAEI competition and it has shown good results compared to the results of other systems in the 2009 competition. In particular, for the anatomy track, its F-Measure of 0.859 tops the best F-Measure achieved in the OAEI 2009 anatomy track. What is more remarkable is that this was achieved with a runtime of only around 2.5 minutes, thereby living up to its name of being an effective and efficient matcher. In addition, its average F-Measure of 0.555 for the conference track ranks second when compared with systems participating in OAEI 2009.

Ontologies	Precision	Recall	F-Measure
cmt-ekaw	0.316	0.545	0.400
conference-edas	0.303	0.588	0.400
conference-iasted	0.350	0.500	0.412
cmt-iasted	0.267	1.000	0.421
edas-iasted	0.471	0.421	0.444
cmt-conference	0.467	0.438	0.452
cmt-confof	0.538	0.438	0.483
conference-ekaw	0.481	0.520	0.500
edas-ekaw	0.500	0.522	0.511
cmt-edas	0.385	0.769	0.513
confof-edas	0.500	0.684	0.578
ekaw-sigkdd	0.538	0.636	0.583
confof-sigkdd	0.667	0.571	0.615
conference-sigkdd	0.588	0.667	0.625
conference-confof	0.550	0.733	0.629
confof-iasted	0.600	0.667	0.632
iasted-sigkdd	0.500	0.867	0.634
ekaw-iasted	0.533	0.800	0.640
edas-sigkdd	0.611	0.733	0.667
confof-ekaw	0.824	0.700	0.757
cmt-sigkdd	0.647	0.917	0.759
Average	0.506	0.653	0.555

Table 3. Results for conference dataset

3.2 Discussions on ways to improve *Eff2Match*

The current implementation of *Eff2Match* only matches concepts and properties with equivalence relations. The techniques used are mainly terminological and structural. Our first proposed improvement to *Eff2Match* is to extend its functionalities to include the discovery of non-equivalence correspondences. Other than the subsumption and disjointedness correspondences defined in OWL, *Eff2Match* will also discover other pre-defined relations that are common within the ontologies to be aligned. For example, in the biomedical domain, many ontologies in the OBO foundry [1] contains the *part-of* relationship but they only connect concepts within the same ontology. We intend to discover relationships like these in a future version of *Eff2Match*.

In addition, the current version of *Eff2Match* requires a few parameters to be set by the user. The tuning of parameters is a manual process which can be tedious and ineffective. Our other proposed improvement to *Eff2Match* is to enable it to tune the parameters automatically, like what has been done in [2].

3.3 Comments on the OAEI 2010 procedure, test cases and measures

In this year's OAEI competition, evaluation was done on the SEALS platform [4] for the benchmark, anatomy and conference tracks. Matchers participating in this track have to

be implemented as web services so that they can be evaluated on the SEALS platform. We feel that this service is very useful for us, particularly for the anatomy track. Unlike the benchmark and conference tracks where the reference alignments are made known to us, evaluation for the anatomy track is done using a blind test. Therefore, it is not possible for us to observe how changes we make to the algorithm affect the results on the anatomy dataset and it is difficult to make improvements. The SEALS platform has alleviated this problem and made evaluation for the anatomy track possible.

Another useful feature of the SEALS evaluation mechanism is that it shows the correct, incorrect and missing correspondences to the participants, allowing them to gain a greater insight of the strengths and weaknesses of their systems. Though this is an extremely useful feature, we feel that the correspondences for the anatomy dataset should not be shown if it were to remain a blind test. The reason is that by joining the set of correct correspondences and the set of missing correspondences, one can easily get hold of the complete reference alignment for the anatomy dataset.

4 Conclusion

We have presented an ontology matcher named *Eff2Match* that can align ontologies efficiently and effectively. Experiments were performed on different pairs of ontologies from three different tracks in OAEI 2010 and results show that *Eff2Match* is able to match real-world ontologies accurately. In addition, it scales well to large ontologies and can be used in applications where the ontology matching process has to be fast.

References

1. The Open Biomedical Ontologies. Available at: `http://obofoundry.org/about.shtml`.
2. Mayssam Sayyadian, Yoonkyong Lee, AnHai Doan, and Arnon Rosenthal. Tuning schema matching software using synthetic scenarios. In *Proceedings of 31st International Conference on Very Large Data Bases (VLDB'05)*, pages 994–1005, Trondheim, Norway, 2005.
3. Nadine Schuurman and Agnieszka Leszczynski. Ontologies for bioinformatics. *Bioinform Biol Insights*, 2:187–200, 2008.
4. SEALS-Semantic Evaluation At Large Scale. Available at: `http://seals.inrialpes.fr/platform`.
5. Semantic Web for Earth and Environmental Terminology (SWEET). Available at: `http://sweet.jpl.nasa.gov/ontology/`.