Results of NBJLM for OAEI 2010

Song Wang[1,2], Gang Wang[1] and Xiaoguang Liu[1]

[1]College of Information Technical Science, Nankai University Nankai-Baidu Joint Lab, Weijin Road 94, Tianjin, China
[2]Military Transportation University, The Equipment support Department, Tianjin, China
jackws66@yahoo.com

Abstract. This paper presents the results obtained by NBJLM (Nankai Baidu Joint Lab Matcher) for its first participation to OAEI 2010. The research of ontology-based similarity calculation among concepts has already been a hot issue. NBJLM is an hybrid ontology alignment method that considers both similarity of literal concept and semantic structure. Simultaneously, how to accelerate matching has been mentioned in this paper and the experimental results show the remarkable improvement of matching speed. In OAEI 2010, NBJLM submitted the result for one alignment task: anatomy.

1 Presentation of NBJLM

In recent years, Ontology matching is mainly used in ontology integration, ontology merging, and ontology reusing. Many approaches to ontology matching have been proposed over the years, references[1][6][4] make full use of information, probability and statistics theory, however, they have limited ability to distinguish semantic differences, and the similarity calculation methods are not perfect. Besides, references[3][5][8][2] have considered various factors, but they do not take into account how to avoid unnecessary calculation to shorten computing time in mapping large-scale ontologies. NBJLM is a multiple strategy dynamic ontology matching system implemented in java. It considers both the literal concept and ontology structure that includes node depth, node density and semantic distance.

1.1 State, purpose, general statement

Given two heterogeneous ontologies O1 and O2, a matching is made up of a set of correspondences between pairs of node IDs belonging to O1 and O2, respectively. NBJLM is designed to find out relations of equivalence and subsumption between entities, i.e. classes and properties, issued from two ontologies. Our approach makes use of the matching strategy that considers literal similarity measure and ontology structure similarity measure. The core contributions of NBJLM is described as followed: Firstly, it uses Hash mapping algorithm to improve efficiency of calculation. Secondly, it takes a full analysis of a number of issues to be considered in structure matching, which makes the algorithm works better, and the matching results are more accurate and efficient. As demonstrated by the experimental results, our method can greatly cut the running time, meanwhile, precise matching results can be obtained.

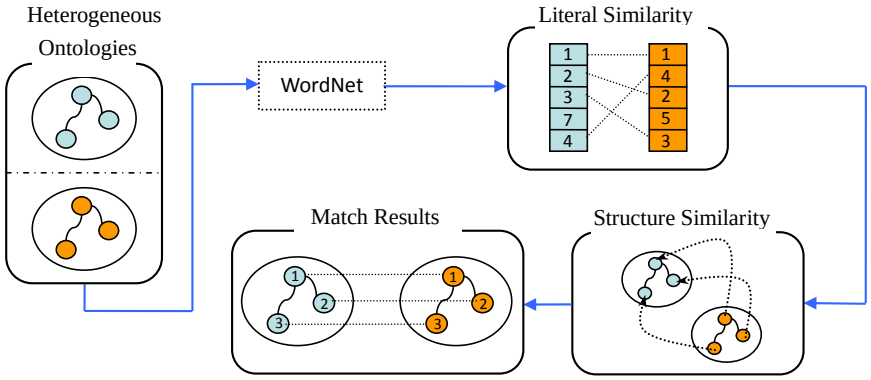

Fig. 1. Procedure of the matching of heterogeneous ontologies.

1.2 Specific techniques used for Anatomy Track

NBJLM uses a new matching strategy that considers literal similarity measure and ontology structure similarity, simultaneously. We obtain the following formula:

$$Sim(ID1, ID2) = \theta \times Sim\_literal(ID1, ID2) + (1 - \theta) \times Sim\_struct(ID1, ID2)$$

where $Sim\_literal(ID1, ID2)$ is the literal concept similarity measure, $Sim\_struct(ID1, ID2)$ is the structural similarity measure, and θ $(0 < \theta < 1)$ is paramater to control how much literal and ontology structure contribute to the ontologies matching respectively. Firstly, the measure of literal similarity is a preliminary matching. It takes account of polysemy and synonym of a word, by transforming the word into a semantic collection using WordNet. Then we can get the preliminary matching results that is semantic mapping rather than spelling mapping of words. Secondly, based on the literal matching results, the measure of ontology structure similarity is calculated through the relation between hypernym and hyponym of a word, considering distance of edges, and depth and density of node in the hierarchy of ontology. With the final combination of the two values, and with adjustment of the parameter, we could obtain more reasonable matching results. The procedure is shown in Fig. 1.

An optimized algorithm for concept sets retrieving If look up a word in WordNet, we can get one or more Synsets (defined by WordNet). For one thing each Synset is a concept set of the words which have the same meaning. For another a word may have several meanings, therefore, each Synset can be used to express one concept of the word. The concept of a node ID in the hierarchy of ontology may be described by several phrases, which are composed of words. That means the concept of the node ID could be described by several Synsets. If we deal with all the Synsets in matching, redundant computation will be inevitable. Therefore, this paper proposes a strategy that obtain the set of Synsets, which are the most similar to the concept of the phrase while

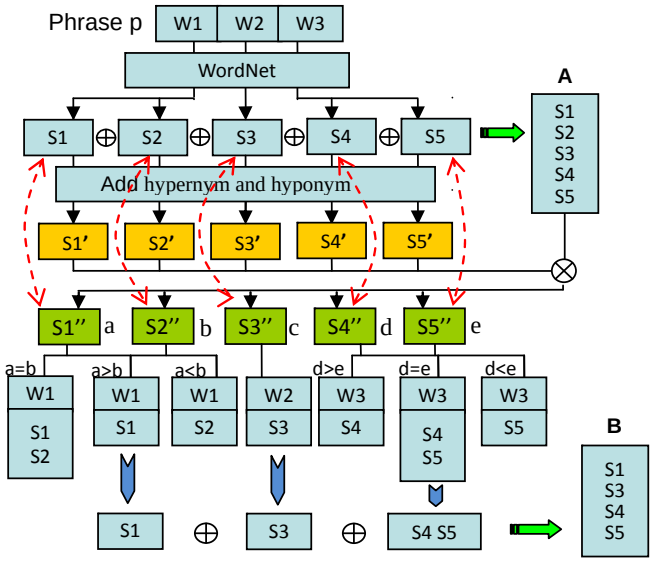

Fig. 2. Get the optimal Synsets to describe the concept of a phrase

include the least Synsets, to describe the concept of a node ID with the help of WordNet. So the unnecessary computation work could be reduced. Fig. 2 describes a simple example that how to tackle a phrase to get the optimal Synsets:

1) Obtain the Synsets of all the words $w1$, $w2$ and $w3$ from phrase p got from a node of $O1$(or $O2$) by WordNet, respectively, $(S1, S2)$, $(S3)$ and $(S4, S5)$.
2) Get the union set of all the Synsets, $A(S1, S2, S3, S4, S5)$, which denotes the concept of phrase p and includes the most Synsets.
3) Add the semantic environment (hypernym and hyponym of the Synset) to Synset. Then we get $S1'$, $S2'$, $S3'$, $S4'$ and $S5'$.
4) Intersect $A(S1, S2, S3, S4, S5)$ with $S1'$, $S2'$, $S3'$, $S4'$ and $S5'$, separately, resulting in $S1''$, $S2''$, $S3''$, $S4''$ and $S5''$. And the numbers of elements of the intersections are a, b, c, d and e. Meanwhile, establish correspondences between $S1$ and $S1''$, $S2$ and $S2''$, $S3$ and $S3''$, $S4$ and $S4''$, $S5$ and $S5''$. The purpose of doing intersections is to find correlation between semantic environment of a Synset and the concept of phrase $\boldsymbol{p}$. The larger number of the intersection's elements is, the more similar relationship between them is.
5) Compare the numbers of intersections' elements mentioned at step 4, which are generated from the same word. And select the Synset of each word, associated with the result of intersection which has the larger number of elements. For example, on the assumption that $a > b$, $d = e$ (c has no comparable object), the Synsets of $w1$, $w2$ and $w3$ are $(S1)$, $(S3)$ and $(S4, S5)$, respectively.
6) Get the union set of $(S1)$, $(S3)$ and $(S4, S5)$, $B(S1, S3, S4, S5)$, which denotes the concept of phrase $\boldsymbol{p}$.

7) It can be found that Synset $S2$ existing in A but not in B is uncorrelated to the concept of phrase p. Therefore, the redundancy can be filtered out by our optimized algorithm. Besides, as increasing in the number of words of phrase, the optimization of the algorithm could be more obvious. Since the matching of nodes in the ontologies is based on the matching of Synsets, the reduction of Synsets, which denote the concepts of nodes in the ontology, will inevitably lead to the reduction of irrelevant semantic mappings and greatly reduce the amount of calculation.

Method of calculation of structural similarity The calculation of structural similarity involves semantic distance with weight, information content, depth and density of node. In order to tackle two ontologies conveniently, we add a virtual common root node which connects two ontologies. So the model could be changed from two independent ontologies to a large ontology, which facilitates the matching. The process of matching is described as follow: firstly, search the common ancestor C of two nodes $c1$ and $c2$. In fact, C is a mapping pair$(c1', c2')$ got from the matching results of literal concepts, where $c1'$ is the ancestral node of $c1$ and $c2'$ is the ancestral node of $c2$. Secondly, calculate the semantic distance between $c1$ and $c2$ through C. Thirdly, do iterative calculation that search the common ancestor C of $c1'$ and $c2'$ until C is the virtual common node. Finally, add depth and density of nodes into the calculation. The formula is:

$$Sim\_struct(ID1, ID2) = Sim\big(Com\_ancestor(c1, c2)\big) \times \frac{\alpha k}{k + Dis(c1, c2)} + \beta\big(\eta + (1-\eta) \times \frac{e(c1) + e(c2)}{2}\big) + \frac{\gamma}{2}\big(\frac{d(c1)}{d(c1)+1} + \frac{d(c2)}{d(c2)+1}\big)$$

Where $Com\_ancestor(c1, c2)$ returns the common ancestor pair of $c1$ and $c2$, and $Dis(c1, c2)$ is the semantic distance, $e(c1)$ and $d(c1)$ are the density and depth of node[8]. The parameters $k(k > 0)$, $\eta(0 < \eta < 1)$, α, β and γ $(\alpha + \beta + \gamma = 1)$ control how much semantic distance, depth, density contribute to the calculation of structural similarity respectively.

$$Dis(c1, c2) = \sum_{x \in pn(c1)} wt\big(x, p(x)\big) + \sum_{x \in pn(c2)} wt\big(p(x), x\big)$$

$$wt(c, x) = Ls(c, x) \times T(c, x)$$

$$Ls(c, x) = -\log\big(P(c|x)\big) = -\log\frac{P(c \bigcap x)}{P(x)} = IC(c) - IC(x)$$

Where $wt(c, x)$ is the weight of $edge(c, x)$, $pn(c)$ is the set of nodes which are on the path from node c to the common ancestor node, $p(x)$ is the parent node of x, $IC(x)$is interest degree[7], $Ls(c, x)$ is the difference of the information content values between a child node and its parent, and $T(c, x)$ is the link relation factor.

There is something important to pay attention to, which makes the algorithm more efficiency:

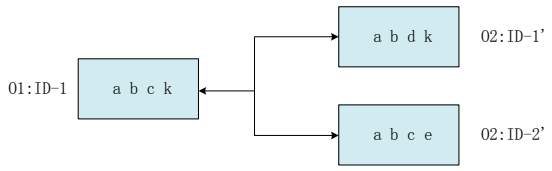

Fig. 3. Literal concept mapping of one to many

- This approach searches all the ancestor nodes of two nodes to be matched, and select the best matching path. If only search the nearest common ancestor node, the result may be wrong. For example: owing to the situation of one to many mappings in the matching results of literal concepts, it may occur that the mappings $(O1 : ID - 2, O2 : ID - 2')$ and $(O1 : ID - 2, O2 : ID - 4')$ got from results of literal concept matching are candidates for structural matching, but in fact $(O1 : ID-2, O2 : ID-2')$ is the best mapping. When comparing the node $O1 : ID-6$ and node $O2 : ID - 6'$, if only search their nearest common ancestor, we will get a pair of nodes, $O1 : ID - 2$ and $O2 : ID - 4$. However, it is not the best mapping pair (we have known that the pair of $O1 : ID - 2$ and $O2 : ID - 2'$ is the best). To avoid this, we need to traverse all the common ancestors of nodes rather than the nearest. Then compare the iterative results and choose the best.
- Involve the literal interest degree. For instance, when we find mapping pairs $(O1 : ID - 1, O2 : ID - 1')$ and $(O1 : ID - 1, O2 : ID - 2')$ have the same structural similarity, and the values of their literal similarity calculations are both $3/4$ as shown in fig. 3, where a, b, c, d, e and k are Synsets, then the literal interest degree is needed to judge which the better matching object of $O1 : ID-1$ from $O2 : ID-1'$ and $O2 : ID - 2'$ is: the less frequency of a Synset occurs in the ontology is, the more it contributes to the meaning of the node. So we calculate all the literal interest degrees of the common Synsets in each mapping pair using the formula metioned in Definition 4. And compare the maximal literal interest degrees of all the mapping pairs, then the max is the best matching because they contain the common Synset whose meaning is closer to concept of the phrase. To suppose the maximal literal interest degree of $(O1 : ID - 1, O2 : ID - 1')$ is $n1$ got from k, simultaneously, the maximal literal interest degree of $(O1 : ID - 1, O2 : ID - 1')$ is $n2$ got from e, and $n1 > n2$, we can draw the conclusion: $(O1 : ID - 1, O2 : ID - 1')$ should be the best mapping pair because $O1 : ID - 1$ is more interested in Synset k.
- At last calculate the factors of density and depth of node. Because in each iteration the value of semantic distance should be multiplied by similarity of the common ancestor node which is smaller than 1, it will surely lead to the similarity of child nodes smaller than those of their ancestor nodes. This is contradictory to the role of depth and density calculation, because the nodes which have greater values of depth and density will have the larger value of similarity. Therefore, we must calculate the depth and density of node out of the procedure of calculation of semantic distance and iterations.

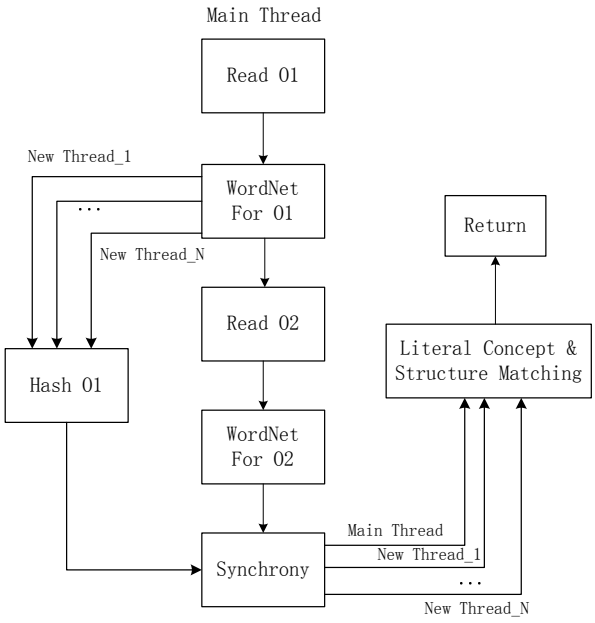

Fig. 4. Parallelization of the algorithm implemented by multi-threads

Parallelization of the algorithm NBJLM uses parallel algorithm to accelerate the process of matching. Fig. 4 shows the task partitioning. Firstly, we use the main thread to read $O1$ file and then look up the Synsets of all the node IDs of $O1$ in the WordNet. The reason of use only one thread is that this stage contains only IO operations which can not benefit from parallel execution and WordNet does not provide thread-safe APIs. Secondly, another multi-threads are launched to calculate hash values of node IDs' Synsets of $O1$, meanwhile we use the main thread to read $O2$ file and look up the Synsets of all the node IDs of $O2$. And these tasks could be run in parallel because one part is CPU operation, and another is IO operation. Finally, we synchronize all the threads, and then use them to calculate the literal concepts similarity and the structure similarity.

1.3 Adaptations made for the evaluation

This year, NBJLM has first taken part in OAEI. Therefore, in OAEI 2010 NBJLM used the match to compute the alignments for one track(anatomy). In order to assure the matching process is fully automated, all parameters are configured automatically with a strategy. No specific adaptations have been made.

1.4 Link to the system and parameters file

The version of NBJLM for OAEI 2010 can be downloaded from our website: $http : //www.brsbox.com/OAEI2010$. The parameter file is also included in the NBJLM.zip

file. I recommend readers to read the readme.txt file first. The file includes the necessary description and parameters as well in brief.

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

NBJLM alignment results for OAEI can be found at
$http://www.brsbox.com/OAEI2010$.

2 Results

In this section, we describe the results of NBJLM algorithm against the Anatomy ontologies provided by the OAEI 2010 campaign. In this test, the real world cases of anatomy for Adult Mouse Anatomy (2744 classes) and NCI Thesaurus (3304 classes) for human anatomy are included. This year we have participated in task#1 for the first time. Experiments were done on a computer with 1.8GHz AMDAthlon dual-core CPU and 2GB DDR2 RAM memory.

2.1 anatomy

Subtrack#1 In this subtrack, participants are asked to maximize F-measure. NBJLM used a threshold equal to 0.8 and obtained an F-measure equal to 85.8%. NBJLM obtained precision equal to 92.0% and recall equal to 80.3%. The runtime was 2 minutes.

3 General comments

3.1 Comments on the results

- **Strengths** NBJLM deals with ontology from two different views and combines results of every step in sequential way. If the ontologies have regular literals and hierarchical structures, NBJLM can achieve satisfactory alignments. And the way of minimizing the comparisons between entities, which leads to enhance running efficiency.
- **Weaknesses** NBJLM depends on the literal concept results to calculate structural similarity. So if the literals of concept missed, NBJLM will get bad results.

3.2 Discussions on the way to improve the proposed system

1) To enrich the semantic dictionaries because WordNet which is not a professional dictionary cannot obtain more comprehensive semantic concepts.
2) To take into account all concepts properties instead of only the hierarchicals ones.

4 Conclusion

This paper reports our first participation in OAEI campaign. We present the alignment process of NBJLM and describe the specific techniques for ontology matching. The method based on heterogeneous ontologies combines the calculations of literal concept and ontology structure and pays more attention to computational efficiency. The strengths and the weaknesses of our proposed approach are summarized and the possible improvement will be made for the system in the future. We propose a brand new algorithm to match ontologies.

References

1. E. Agirre and G. Rigau. A proposal for word sense disambiguation using conceptual distance. *AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE SERIES 4*, pages 161–172, 1997.
2. B. L. X. H. Y. L. K. J. Tang, J. Li. Using Bayesian decision for ontology mapping. *Journal of Web Semantics: Science, Services and Agents on the WorldWideWeb*, pages 243–262, 2006.
3. Y. Jean-Mary, E. Shironoshita, and M. Kabuka. Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, page 235C251, 2009.
4. Y. Li, D. McLean, Z. Bandar, J. O'Shea, and K. Crockett. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138–1150, 2006.
5. M. A. Q. Muhammad Fahad. Similarity Computation by Ontology Merging System: DKP-OM. *Computer, Control and Communication*, pages 17–18, February 2009.
6. P. Resnik et al. Using information content to evaluate semantic similarity in a taxonomy. In *International Joint Conference on Artificial Intelligence*, volume 14, pages 448–453. Citeseer, 1995.
7. S. Ross. A first course in probability. *New York*, 1994.
8. J. Sevilla, V. Segura, A. Podhorski, E. Guruceaga, J. Mato, L. Martinez-Cruz, F. Corrales, and A. Rubio. Correlation between gene expression and GO semantic similarity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 2(4):338, 2005.