Neural Network based Constraint Satisfaction in Ontology Mapping

Ming Mao*
SAP Research
Palo Alto, CA 94304 USA
ming.mao@sap.com

Yefei Peng*
Yahoo!
Sunnyvale, CA 94089 USA
ypeng@yahoo-inc.com

Michael Spring
University of Pittsburgh
Pittsburgh, PA 15260 USA
spring@sis.pitt.edu

Abstract

Ontology mapping seeks to find semantic correspondences between similar elements of different ontologies. Ontology mapping is critical to achieve semantic interoperability in the WWW. Due to the fact that ubiquitous constraints (e.g., hierarchical restrictions in RDFS) caused by the characteristics of ontologies and their representations exist in ontologies, constraints satisfaction has become an intriguing research problem in ontology mapping area. Though different techniques have been examined to find mappings, little work is made to solve constraint satisfaction problem for ontology mapping. Currently most approaches simply validate ontology constraints using isolate heuristic rules instead of comprehensively considering them in a global view. This paper proposes a neural network based approach to search for a global optimal solution that can satisfy ontology constraints as many as possible. Experimental results on OAEI benchmark tests #248-#266 show the approach is promising. It dramatically improves the performance of preliminary mapping results.

1. Introduction

The World Wide Web (WWW) is widely used as a universal medium for information exchange. However, semantic interoperability in the WWW is still limited due to the heterogeneity of information. Ontology, a formal, explicit specification of a shared conceptualization (Gruber 1993), has been suggested as a way to solve the problem. With the popularity of ontologies, *ontology mapping*, aiming to find semantic correspondences between similar elements of different ontologies, has attracted many research attentions from various domains. Different techniques have been examined in ontology mapping, e.g., analyzing linguistic information of elements in ontologies (Qu, Hu *et al.* 2006), treating ontologies as structural graphs (Melnik, Garcia-Molina *et al.* 2002), using heuristic rules (Hovy 1998) or applying machine learning techniques (Doan 2002). Comprehensive surveys of the state of the art ontology mapping approaches can be found in (Euzenat, Bach *et al.* 2004; Kalfoglou and Schorlemmer 2003; Noy 2005; Euzenat and Shvaiko 2007).

Though the state of the art ontology mapping approaches have made significant progress in finding mappings, few of them comprehensively consider ontology constraints so as to further improve mapping accuracy. In fact, most approaches simply validate ontology constraints using isolate heuristic rules instead of comprehensively considering them in a global view.

This paper proposes a neural network based approach to search for a global optimal solution that can satisfy ontology constraints as many as possible. Experimental results on OAEI benchmark tests[1] show the approach is promising. It dramatically improves the performance of preliminary mapping results.

2. A Simple Scenario of Constraint Satisfaction in Ontology Mapping

Constraint satisfaction has become an intriguing problem in ontology mapping area due to the ubiquity of ontology constraints. For example, the hierarchical relations in RDFS[2] do not allow crisscross mappings, the axioms such as *owl:sameAs* and *owl:equvalentClass* in OWL[3] indicate an equivalent relation between different elements, and the rules in SWRL[4] would be to imply or assert some properties that are not directly available.

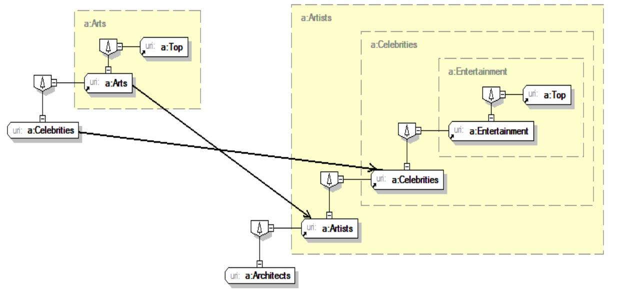

Figure 1. A simple scenario of constraint satisfaction in ontology mapping

Figure 1 is a very simple real world ontology mapping case between the *taxonomy* of Google directory[5] and Yahoo

* This work was done when the authors had affiliations with University of Pittsburgh.

[1] http://oaei.ontologymatching.org/2007/benchmarks/
[2] http://www.w3.org/TR/rdf-schema/
[3] http://www.w3.org/TR/owl-features/
[4] http://www.daml.org/2003/11/swrl/
[5] http://directory.google.com/

directory[6]. In the example, a crisscross mapping (i.e., *celebrities* maps to *celebrities* & *arts* maps to *artists*) might be incorrectly output as final results based on the lexical analysis of the ontologies only. To eliminate such kind of invalid mappings and thus improve the quality of ontology mapping, it is critical to find an optimal configuration that can best satisfy ontology constraints, such as "if e_{1i} *maps to* e_{2j} is true, then e'_{1i} *maps to* e'_{2j} is false, where e_{xy} is an element (e.g., class or property) in ontology O_x, e'_{1i} is the parent of e_{1i}, and e'_{2j} is the child of e_{2j}.".

Finding a configuration to satisfy the constraints as many as possible is known as constraint satisfaction problem (CSP). CSPs are typically solved by a form of search, e.g. backtracking, constraint propagation or local search (Tsang 1993). In 1981, McClelland and Rumelhart proposed to use the interactive activation and competition (IAC) neural network to solve CSPs in word perception (McClelland and Rumelhart 1981). In this paper, we briefly introduce the mechanism of the IAC neural network. A comprehensive introduction of the network can be found in (McClelland and Rumelhart 1988).

3. Our Approach

Before we discourse the IAC neural network model, we talk a little bit about the whole procedural of our approach. Given a mapping task, i.e., the OAEI benchmark tests #248-#266, first, we parse ontologies using Jena[7], and preprocess them by removing stop words, stemming, and tokenizing. Next, we measure three kinds of similarities, i.e., edit distance based similarity, profile similarity and structure similarity, for each ontology. After that we calculate the *harmony* for each similarity by counting the normalized number of mapping pairs that suggest unambiguously 1-to-1 mappings, and then we adaptively aggregate three similarities upon their harmonies. Based on the aggregated similarity, we activate the IAC neural network to search for an optimal configuration that best satisfy ontology constraints. Finally we extract mapping results using naive descendant extraction algorithm (Meilicke and Stuckenschmidt 2007). This paper focuses on the implementation of the IAC network in the context of ontology mapping. We refer interested readers to (Mao and Peng 2006; Mao, Peng *et al.* 2007) for details about how we measure and aggregate three similarities to generate preliminary mapping results. However it is worth to remark that our neural network based constraint satisfaction approach is a generic approach, and thus the preliminary results can come from any similarity-based ontology mapping approach, not limited to ours.

3.1. The IAC Neural Network

Generally speaking, an IAC neural network consists of a number of competitive *nodes* that are connected to each other. Each node represents a *hypothesis*. The *connection* between two nodes represents *constraint* between their hypotheses. If two hypotheses support each other, the connection between them is *positive* (i.e., *active*); whereas if two hypotheses are against each other, the connection between them is *negative* (i.e., *competitive*). Each connection is associated with a *weight*, which is proportional to the strength of the constraint. The activation of a node is determined locally by four sources: its initial activation, the input from its adjacent nodes, its bias and some external input. The mechanism of the IAC neural network can be illustrated using the following simple example.

Suppose we have two grids, 1 and 2, and two constraints:

1. Each grid can have one value, either A or B.
2. The values of the two grids are different.

We also have four hypotheses: A in grid 1 (H_{A1}); B in grid 1 (H_{B1}); A in grid 2 (H_{A2}); B in grid 2 (H_{B2}). Based on the two constraints we know there are two negative connections and one positive connection for each hypothesis.

1. H_{Ai} is *against* H_{Bi}, and vice versa (i=*1* or *2*)
2. H_{x1} is *against* H_{x2}, and vice versa (x=*A* or *B*)
3. H_{Ai} *supports* H_{Bj}, and vice versa (i, j = *1* or *2*, and $i \neq j$)

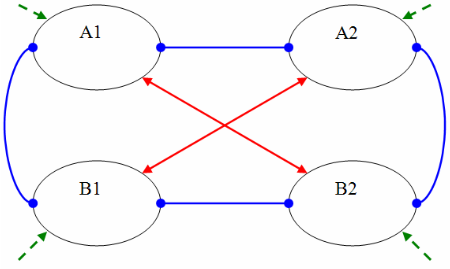

Figure 2. A simple IAC neural network

Figure 2 illustrates the simple example, in which each node represents a hypothesis, the line with rounded head and arrowhead represents negative connection and positive connection between hypothesis respectively and the dashed line with arrowhead represents a small stimulus on each node from outside. Assume the negative weight is half the positive weight and all nodes are inactive at start. Though the input from three neighbors of the node will cancel out, the small excitatory input from outside will activate the node. All nodes are chosen to be updated sequentially in random order. Finally, either *A1 & B2* or *B1 & A2* will be active, and finally the network will reach a *stable state*.

[6] http://dir.yahoo.com/
[7] http://jena.sourceforge.net/

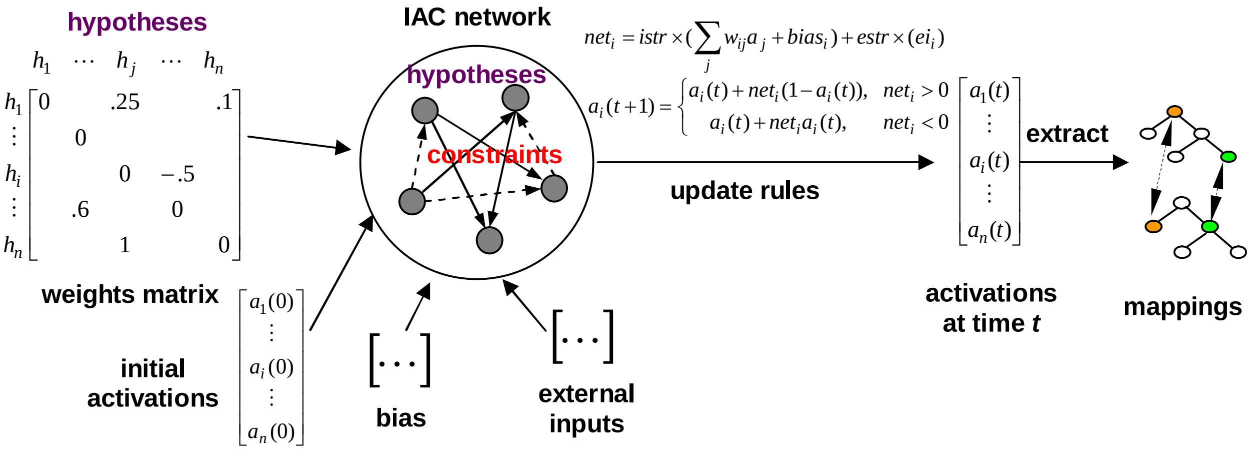
hypotheses
$h_1 \cdots h_j \cdots h_n$
h_1 0 .25 .1
0
h_i 0 −.5
.6 0
h_n 1 0
weights matrix
initial activations
$a_1(0)$ ⋮ $a_i(0)$ ⋮ $a_n(0)$
IAC network
hypotheses
constraints
bias
external inputs
$net_i = istr \times (\sum_j w_{ij} a_j + bias_i) + estr \times (ei_i)$
$a_i(t+1) = \begin{cases} a_i(t) + net_i(1 - a_i(t)), & net_i > 0 \\ a_i(t) + net_i a_i(t), & net_i < 0 \end{cases}$
update rules
$a_1(t)$ ⋮ $a_i(t)$ ⋮ $a_n(t)$
activations at time *t*
extract
mappings

Figure 3. The IAC neural network in the context of ontology mapping

3.2. The Motivation

The common properties between the characteristic of ontology mapping problem and the mechanism of the IAC network motivate the work addressed in the paper. First, in ontology mapping, the constraints between mapping hypotheses are either *active* or *competitive*. For example, given two mapping hypotheses, e.g., e_{1i} maps to e_{2j} and e'_{1i} maps to e'_{2j}, the constraint "if e_{1i} maps to e_{2j} is true, then e'_{1i} maps to e'_{2j} is true, where e'_{1i} and e'_{2j} are the child of e_{1i}, and e_{2j} respectively" is active; whereas the constraint "if e_{1i} maps to e_{2j} is true, then e'_{1i} maps to e'_{2j} is false, where e'_{1i} is the parent of e_{1i}, and e'_{2j} is the children of e_{2j}" is competitive. Such characteristic is exactly the same as that in the IAC networks. Secondly, preliminary mapping results can bring prior knowledge to us, which makes the IAC neural network more practical. That is, the aggregated similarity of each mapping pair just reflects the confidence of the mapping hypothesis, and thus it can be directly used as initial activation or be converted as external inputs or bias of a node in the IAC network.

3.3. The Implementation

Figure 3 illustrates the implementation of IAC neural network in the context of ontology mapping, where a_i denotes the activation of *node i*, written as n_i, net_i denotes the net input of the node, *istr* and *estr* denote the parameter of internal and external input respectively, the w_{ij} denotes the connection weight between n_i and n_j, a_j denotes the activation of node n_j, $bias_i$ denotes the bias of n_i, and ei_i denotes the external input of n_i, which is a function of the confidence of a mapping. In the picture, a node (e_{1i}, e_{2j}) represents a hypothesis that indicates a mapping between e_{1i} and e_{2j}. The *connections* between nodes represent *constraints* between hypotheses. For example, the constraint that "only 1-to-1 mapping is allowed" results in a *negative* connection between nodes (e_{1i}, e_{2j}) and (e_{1i}, e_{2k}), where $k \neq j$. Moreover, "two elements match if their children match", results in a *positive* connection between nodes (e_{1i}, e_{2j}) and (e_{1k}, e_{2t}), where e_{1k} and e_{2t} are the children of e_{1i} and e_{2j} respectively. Currently we implemented 12 constraints (see Table 1). The *weights* in weight matrix correspond to the prior confidence of the constraint, which are currently set as 1 for positive constraints and -1 for negative constraints. The *initial activation* of each node is set to the *aggregated similarity* of (e_{1i} , e_{2j}) from previous processes. The *bias* of each node is set as 0. The *external input* is set to the reliability of each hypothesis. Currently the external input of unambiguous hypotheses, which hold the highest similarity in its responding row and column in the similarity matrix, is set as 10, otherwise as 0. The activation of a node can be updated with the rule illustrated in the picture, where a_i denotes the activation of *node i*, written as n_i, net_i denotes the net input of the node. Once the network starts running, it can either be stopped after *n* cycles or at some *goodness* point, which is the degree of how many desired constraints are satisfied, at time *t* (McClelland and Rumelhart 1988). In our implementation, we let the network stop when its *delta goodness* reaches some satisfaction (i.e., ≤1%). Finally, please note, in Table 2, though the number of negative constraints is much less than the number of positive constraints, the ratio of negative connections and positive connections is not small due to the fact that each node in the network will have a large amount of negative connections introduced by the constraint that "only 1-1 mapping is allowed".

4. Evaluation

4.1. Data Sets

To evaluate our approach we use the benchmark tests #248-#266 from OAEI ontology matching campaign 2007[8]. The tests include 1 reference ontology, which describes the very narrow domain of bibliography, and 15

[8] http://oaei.ontologymatching.org/2007/

Table 1. The constraints used in our approach

#	Constraints	Connection
1	Only 1-1 mapping is allowed.	negative
2	No crisscross mapping is allowed.	negative
3	If children elements match, then their parent elements match.	positive
4	If parent elements match, then their children elements match.	positive
5	If e_{1i} match e_{2j}, then e_{1s} match e_{2t}, where e_{1i} and e_{1s}, e_{2j} and e_{2t} are siblings in ontologies.	positive
6	If property elements match, then their domain elements match.	positive
7	If property elements match, then their range elements match.	positive
8	If class elements match, then their direct property elements match.	positive
9	If property elements match, then their mother-class elements match.	positive
10	If class elements match, then their individual elements match.	positive
11	If individual elements match, then their mother-class elements match.	positive
12	Two elements match if their *owl:sameAs* or *owl:equivalentClass* or *owl:equivalentProperty* elements match.	positive

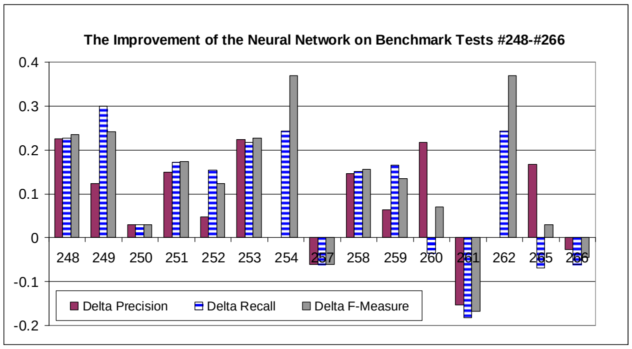

Figure 4. Results of the IAC neural network approach on OAEI benchmark tests #248-#266

Table 2. The overall improvement of our approach on #248-#266

H-Mean	Precision	Recall	F-Measure
Before NN	.76	.54	.63
After NN	.88	.67	.76
NN Improvement	13%	24%	19%

artificially-made test ontologies, each of which discards various information from the reference ontology so as to evaluate how algorithms behave when some kind of information is lacking.

The reasons why we choose OAEI benchmark tests #248-#266 are: 1.The OAEI benchmark tests have become authoritative tests in the area of ontology mapping. 2. The ground truth of the benchmark tests is open, and thus can be used for comprehensive evaluation. 3. Tests #248-#266 are the most difficult tests among all benchmark tests. The results from all participants[9] on these tests are pretty lower than their results on other benchmark tests. Therefore the improvement on these tests can greatly contribute to the overall performance of all kinds of ontology mapping approaches.

4.2. Evaluation Criteria

We follow the evaluation criteria from the OAEI campaign, calculating the *precision* (i.e., the ratio of *correctly found mappings* to *all found mappings*), *recall* (i.e., the ratio of *correctly found mappings* to *all true positive mappings*) and *f-measure* (i.e., the weighted harmonic means) over each benchmark test (Euzenat *et al.* 2007).

4.3. Experimental Methodology and Results

The experiment methodology is: Given the preliminary mapping results, we activate the IAC neural network on each OAEI benchmark test in #248-#266. The value of parameter in the network is set as what we described in §3.3. Then we let the network run by itself and stop it when its *delta goodness* is less than 1%. Finally we extract mapping results using naïve descendant extraction algorithm (Meilicke and Stuckenschmidt 2007).

Experiment results in Figure 4 show the neural network based constraint satisfaction approach improves the *f-measure* of 12 tests among 15 tests except #257, #261 and #266. The largest improvement of f-measure (i.e., .37) happens on #254 and #262. The decrease on #261 is due to the extension of its structure, i.e., new classes are added as new layers in test ontology, which makes some constraints in neural network are not correct anymore. Meanwhile, no linguistic information is available in #261 at all, and thus there is no linguistic analysis that we can rely on. Table 2 shows the prominent improvement of our approach over *all* tests #248-#266. They are 13%, 24%, and 19% for *precision*, *recall*, and *f-measure* respectively.

5. Related Work

Different approaches have been proposed to solve the ontology mapping problem. In this section we only review three most related work, i.e., GLUE (Doan 2002), Similarity Flooding (Melnik, Garcia-Molina *et al.* 2002), and Falcon-AO (Qu, Hu *et al.* 2006), from the perspective of constrain satisfaction. Comprehensive surveys of some famous ontology mapping systems, such as COMA (Do and Rahm 2002), QOM (Ehrig and Staab 2004) and

[9] http://oaei.ontologymatching.org/2007/results/benchmarks/HTML/results.html

PROMPT (Noy and Musen 2003), can be found in (Euzenat, Bach *et al.* 2004; Kalfoglou and Schorlemmer 2003; Noy 2005; Euzenat and Shvaiko 2007).

GLUE (Doan 2002) is an instance based ontology (specifically taxonomy) mapping approach, which implements multiple learning based matchers to exploit information in concepts/instances and taxonomic structure of ontologies. From constraint satisfaction view, GLUE adopts relaxation labeling approach to search for the mapping configuration that best satisfies the domain constraints. GLUE and our approach are similar in that we both try to comprehensively consider ontology constraints by utilizing fledged techniques from other domains. However, we explore different approaches and different kinds of constraints. GLUE adopts relaxation labeling approach that has been applied successfully in computer vision (Hummel and Zucker 1983), natural language processing (Padro 1998) and hypertext classification (Chakrabarti, Dom *et al.* 1998); whereas our approach integrates the IAC neural network that has been used to model visual word recognition (McClelland and Rumelhart 1981) and information retrieval tasks (Ross and Philip 1991). Furthermore, the constraints implemented in our approach are more complex than that in GLUE. We consider 12 constraints from hierarchical relations to OWL axioms, all of which are general constraints and independent to specific domain. Whereas, GLUE explores eight constraints, and four of them are specific and domain-dependent. The other four constraints, though general, are about simple hierarchical relations only. Finally, the relaxation labeling approach in GLUE needs to estimate prior probability of data distribution, which relies on the availability of instances in taxonomies. Unfortunately large number of instances usually is not available in most ontology mapping cases. Though our neural network needs prior confidence to set parameters, the similarity from preliminary mapping results just reflects such confidence in some degree.

Similarity Flooding (Melnik, Garcia-Molina *et al.* 2002) is a generic graph based ontology mapping approach. It utilizes fixpoint computation to determine corresponding nodes in the graphs. The principle of the similarity flooding (SF) approach is that the similarity between two nodes depends on the similarity between their adjacent nodes. The most likeness between us is both of us utilize graph theory either directly or indirectly when satisfying ontology constraints. The SF employs dependency graph, and we convert graphic relations into some rules that neural network can take into account. However we are different in several ways. First, the dependency graph in SF does not support competitive constraints. Instead, our neural network model can deal with both active and competitive constraints. Furthermore, the dependency graph only works for directed labeled graphs. When labeling is uniform or undirected, or when nodes are less distinguishable, SF degrades. Contrarily, our neural network approach is flexible on this aspect. Finally, the number and the complexity of the constraints implemented in SF are much less than ours.

Falcon-AO (Qu, Hu *et al.* 2006) is a similarity-based generic ontology mapping system. It consists of three linguistic matchers, one structure matcher, and one ontology partitioner. The common place between us is both approaches are measuring multiple similarities especially when generating linguistic similarity the concept of our *profile* is very similar as their *virtual document*. However, we are different in many ways. From the constraint satisfaction view, Falcon-AO uses a bipartite graph when measuring structural similarity of two ontologies. However bipartite graph can only explore simple hierarchical relations instead of complex ontology axioms or rules. Meanwhile, though Falcon-AO forms three heuristic rules to integrate multiple similarities, they do not have any solutions to further optimize final results so as to satisfy various ontology constraints. Therefore, Falcon-AO can be seen as the first step of our approach. That is, we can adopt its results as our preliminary results for further validation.

6. Conclusion and Future Work

In this paper we proposed an IAC neural network based approach to find a global optimal solution that best satisfies ontology constraints. The experimental results show the approach dramatically improves the performance of preliminary mapping results on OAEI benchmark tests #248-#266. Future work may include exploring more complex constraints, optimizing weight matrix, implementing the neural network in parallel computing platforms to improve its efficiency etc.

7. References

S. Chakrabarti, B. Dom, and P. Indyk. Enhanced Hypertext Categorization Using Hyperlinks. In Proceedings of the ACM SIGMOD Conference, 1998.

Do, H.H., and Rahm, E. COMA - A System for Flexible Combination of Schema Matching Approaches. VLDB 2002, 610-621

Doan, A., J. Madhaven, *et al.* "Learning to Match Ontologies on the Semantic Web." VLDB Journal **12**(4): 303-319. 2002.

Ehrig, M., and Staab, S. QOM - Quick Ontology Mapping. International Semantic Web Conference 2004, 683-697

Euzenat, J., Bach, T., *et al.* State of the art on ontology alignment, Knowledge web NoE. 2004.

Euzenat, J *et al.* Results of the Ontology Alignment Evaluation Initiative 2007. In Proceedings of ISWC 2007 Ontology Matching Workshop. Busan, Korea. 2007.

Jerome Euzenat and Pavel Shvaiko. Ontology Matching. Springer-Verlag. Heidelberg (DE). 2007.

Gruber, T. (1993). "A Translation Approach to Portable Ontology Specifications." Knowledge Acquisition **5**(2): 199-220

Hovy, E. Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC), Granada, Spain. 1998.

R. Hummel and S. Zucker. On the Foundations of Relaxation Labeling Processes. PAMI, 5(3):267-287, May 1983.

Kalfoglou, Y. and M. Schorlemmer "Ontology mapping: the state of the art." Knowledge Engineering Review **18**(1):1-31. 2003.

Mao, M. and Peng, Y. The PRIOR: Results for OAEI 2006. In Proceedings of ISWC 2006 Ontology Matching Workshop. Atlanta, GA. 2006

Mao, M., Peng, Y. and Spring, M. A Profile Propagation and Information Retrieval Based Ontology Mapping Approach, In Proceedings of SKG 2007.

J. McClelland and D. E. Rumelhart, "An interactive activation model of context effects in letter perception: part 1. An account of basic findings," *Psychological Review*, vol. 88, pp. 375-407, 1981.

J. McClelland and D. Rumelhart. Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises. The MIT Press. 1988

C. Meilicke, H. Stuckenschmidt. Analyzing Mapping Extraction Approaches. In Proceedings of ISWC 2007 Ontology Matching Workshop. Busan, Korea.

Melnik, S., H. Garcia-Molina, *et al.* Similarity flooding: a versatile graph matching algorithm and its application to schema matching. Proc. 18th International Conference on Data Engineering (ICDE). 2002.

Noy, N. and Musen, M. The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping. International Journal of Human-Computer Studies 59: 983-1024

Noy, N. "Semantic Integration: A Survey of Ontology-Based Approaches." SIGMOD Record **33**(4): 65-70.

L. Padro. A Hybrid Environment for Syntax-Semantic Tagging, 1998.

Qu, Y., Hu, W., and Cheng, G. Constructing virtual documents for ontology matching. In Proceedings of the 15th International Conference on World Wide Web. 2006.

Ross, W. and Philip, H. "Using the cosine measure in a neural network for document retrieval," in *Proceedings of the 14th SIGIR Conference*. Chicago, IL. 1991.

E. Tsang, *Foundations of Constraint Satisfaction*: Academic Press, 1993.