ABSURDIST II: A Graph Matching Algorithm and its Application to Conceptual System Translation

Ying Feng
Computer Science Department,
Indiana University, Bloomington, IN
yingfeng@cs.indiana.edu

Robert L. Goldstone
Psychology Department,
Indiana University, Bloomington, IN
rgoldsto@indiana.edu

Vladimir Menkov
Aqsaqal Enterprises,
Penticton, BC
vmenkov@cs.indiana.edu

Abstract

ABSURDIST II, an extension to ABSURDIST, is an algorithm using attributed graph matching to find translations between conceptual systems. It uses information about the internal structure of systems by itself, or in combination with external information about concept similarities across systems. It supports systems with multiple types of weighted or unweighted, directed or undirected relations between concepts. The algorithm exploits graph sparsity to improve computational efficiency. We present the results of experiments with a number of conceptual systems, including artificially constructed random graphs with introduced distortions.

Introduction

The problem of translating between conceptual systems is of substantial interest in cognitive science. It asks the following question: given two conceptual systems A and B, each one consisting of some concepts and relations between them, how can correspondences across the systems be established? A general system for translating between conceptual systems would be valuable in many domains. We would like a method for translating between two individuals who have been acculturated with different languages or terminologies, between two time-slices of the same individual to track their cognitive development, between two scientific communities committed to fundamentally different theoretical ontologies to determine potential continuities across historical change, between related databases using different XML representations, and between different knowledge structures within the same individual to identify useful analogies for induction.

There are two major strategies to establish corresponding concepts across conceptual systems. The first approach of External Grounding establishes correspondences by finding common external referents for conceptual elements across two systems. The second approach uses purely internal information within the systems to establish correspondences. John's and Jane's **Cat** concepts can potentially be placed into correspondence with each other because they play similar roles within their respective conceptual networks, including matching **is-a** relations to **Animal**, **eats** relations to **Friskies**, and **has-a** relations to **Paw**. Some argue that this account is hopelessly circular (Fodor, 1998). Jane's and John's **Cat** concept can only be placed into correspondence if one already knows that their **Friskies** concepts match, and this is just as much in need of explanation as how to match **Cat** concepts. However, other researchers (Lenat & Feigenbaum 1991) have argued that the circularity, while certainly present, is not hopeless. With proper algorithms, cross-system correspondences can mutually and simultaneously constrain one another (Goldstone & Rogosky 2002). The current work explores a method for integrating internal and external determinants of conceptual translation.

A conceptual system can be formalized as a *directed weighted labeled graph* (discussed below), also known as an *attributed graph*. Thus, matching two conceptual systems can be formalized as finding a match between two attributed graphs and evaluating the quality of this match.

Graph matching is a mature computer science research area. While the original research focus was on finding exact isomorphisms between two graphs, a substantial amount of work has been done on finding approximate, or "error-correcting" isomorphisms as well. Those latter efforts are directed at finding, for two graphs G_1 and G_2, an isomorphism $s(\cdot)$ such that the "distance" between $s(G_1) = S$ and G_2 is minimized. The definition of distance is application specific. It can be based, for example, on the number of editing operations needed to convert S into G_2, or on the Euclidean norm of the difference between the matrices describing S and G_2.

A detailed bibliography on the approximate isomorphism problem can be found in (Messmer 1995). There have been three main lines of research. First, the classic structure matching algorithms first developed for finding exact isomorphism can be extended to the case of approximate matching. Messmer's own work is in this tradition. Second, one can approach the approximate isomorphism problem as the problem of minimizing a certain error function on the space of correspondence matrices subject to structural constraints, and solve the problem using the full arsenal of optimization methods (Gold & Rangarajan 1996; Rangarajan & Mjolsness 1994; Pelillo 1998). Third, a number of neural-net and related iterative approaches have been presented (Schadler & Wysotzki 1997; Schädler & Wysotzki 1999; Melnik, Garcia-Molina, & Rahm 2002).

This paper continues work on the simple approach of (Goldstone & Rogosky 2002) that was originally inspired by constraint propagation neural networks for consistent visual interpretation or analogical reasoning. We begin with graph representation of conceptual systems, and an extended and improved translation algorithm based on ABSURDIST, using only internal, within-system information. Then we present experiments showing interaction of internal and external sources of information when both are available.

Graph Representation and Matching Algorithm

ABSURDIST II is based on the ABSURDIST algorithm described in (Goldstone & Rogosky 2002). The essential process in the algorithm creates a correspondence matrix of possible cross-system translations, updating it at each iteration using a net input including external similarity, excitation, and inhibition components. After a fixed number of iterations, the values in the correspondence matrix are used to select the mapping between the two graphs. ABSURDIST II introduces a number of improvements over the original algorithm: the input representations can be arbitrary combinations of binary relations; sparsity in the systems' representations is exploited to decrease computational complexity; the overall activity in correspondence units is dynamically adjusted to assure appropriate patterns of connectivity across systems; a post-iteration process is introduced to assure a 1-to-1 mapping selection based on the correspondence matrix.

Graph Representation of Conceptual Systems

A conceptual system with N concepts $\{A_1, \ldots, A_N\}$ can be represented as an $N \times N$ matrix $G \in (\{0\} \cup S)^{N \times N}$. In this matrix each element $g_{ij} \in (\{0\} \cup S)$ describes the totality of the relations existing between concepts A_i and A_j. The set S is chosen to be suitable to represent all possible non-empty combinations of relations that may exist between a pair of nodes. (The value 0 is used to describe the absence of any relations between the two concepts).

Alternatively, the conceptual system can be represented as a directed graph with N concepts, with each concept represented by a node. Concepts are connected by edges carrying *labels* from the same set S. Its value is the same as the value of the matrix element g_{ij}, and describes the totality of the relations between concepts A_i and A_j.

Depending on the complexity of the conceptual system, various types of graph representations can be chosen. First, consider a system with only binary valued relations between the concepts. Such a system can be described by an undirected or directed graph with no explicitly stored labels associated with edges.

In more complex systems, the relation may have some kind of strength or weight associated with it. For example, the **Cooccurrence** relation in a conceptual system describing the vocabulary of a corpus may have a weight corresponding to the degree of cooccurrence of the terms it links. In some systems based on human perception of facts, a relation may have a probability- or likelihood-related weight. For such a system we would want to use a graph where real-valued labels are associated with the edges.

Finally, in an even more general case when M possibly weighted relation types $R_1, R_2, \ldots, R_M$ exist in the system, the system can be represented by a directed graph whose labels are vectors from the M-dimensional space $S = \mathrm{R}^M$. We use the notation $\vec{a}_{qr}$ for the label associated with the edge going from A_q to A_r in the graph. The i-th component of this vector, a^i_{qr}, is the weight of the relation R_i on this edge. If no edge goes from A_q to A_r, then the value of $\vec{a}_{qr}$ will be a zero vector, $\vec{a}_{qr} = \vec{0}$.

For convenience, our implementation of ABSURDIST II assumes that all relations are directed. If the conceptual system involves an undirected (symmetric) relation, such as **Similar-to**, existing between concepts A_1 and A_2, then we simply store it twice, as **Similar-to**(A_1, A_2) and **Similar-to**(A_2, A_1). We also assume that all relation weights are in the $[0;1]$ range, which means that all edge labels in fact belong to $[0;1]^M$.

The original ABSURDIST algorithm uses the concept of "similarity of psychological distances" $S(D(A_q, A_r), D(B_x, B_y))$ which measures how similar relation between the concepts A_q and A_r in system A is to the relation between the concepts B_x and B_y in system B. To extend it to directed graphs with labels from $[0;1]^M$, we first define the *directed edge similarity function* $S_{\mathrm{d}}(\vec{a}_{qr}, \vec{a}_{xy})$ as

$$S_{\mathrm{d}}(\vec{a}_{qr}, \vec{a}_{xy}) = 1 - D_{\mathrm{d}}(\vec{a}_{qr}, \vec{b}_{xy}). \quad (1)$$

The *directed edge difference function* $D_{\mathrm{d}}(\vec{a}, \vec{b})$ used above is simply the normalized 1-norm of the vector difference of the edge labels, viz.

$$D_{\mathrm{d}}(\vec{a}_{qr}, \vec{b}_{xy}) = \|\vec{a}_{qr} - \vec{b}_{xy}\|_1 = \frac{1}{M}\sum_{i=1}^{M} |a^i_{qr} - b^i_{xy}|. \quad (2)$$

It is possible, of course, to define the edge difference using the 2-norm or the ∞-norm of the vector difference, instead of the 1-norm.

To compute the excitation matrix in ABSURDIST II, we will use, in place of the "similarity of psychological distances", *undirected, or symmetrized, edge similarity*:

$$S(\vec{a}_{qr}, \vec{a}_{xy}) = 1 - D(\vec{a}_{qr}, \vec{a}_{xy}) \quad (3)$$

with $D(\vec{a}_{qr}, \vec{a}_{xy}) = D(\vec{a}_{rq}, \vec{a}_{yx})$ defined via the symmetrized formula:

$$D(\vec{a}_{qr}, \vec{a}_{xy}) = \frac{1}{2}(D_{\mathrm{d}}(\vec{a}_{qr}, \vec{a}_{xy}) + D_{\mathrm{d}}(\vec{a}_{rq}, \vec{a}_{yx})). \quad (4)$$

Exploiting Sparsity to Reduce Computational Complexity

At every iteration step in the original ABSURDIST algorithm, the excitation $R(A_q, B_x)$ is computed for each concept pair as follows: $R(A_q, B_x) = \sum_{r \neq q} \sum_{y \neq x} S(D(A_q, A_r), D(B_q, B_y)) C(A_r, B_y)/(N - 1)$. Computing the N^2 values of $R(\cdot, \cdot)$ by this formula requires $O(N^4)$ operations at each iteration. Fortunately, the matrix or graph describing a typical conceptual system is rather sparse. The average degree D of a node is much smaller than N. Moreover, in many domains, as N grows, D does not grow as fast as $O(N)$. We will show how to use this sparsity to carry out each iteration in $O(N^2D^2)$ time instead of $O(N^4)$.

For simplicity, we will use the symbol A to refer to the set of all subscripts $\{1, 2, \ldots, N_A\}$, as well as to the conceptual systems A itself, and use the following notation:

$$C(S_1, S_2) = \sum_{q \in S_1} \sum_{r \in S_2} C(A_q, B_r);$$

$\Omega(q) =$ set of all indices of all nodes of A connected to A_q;

$$\Omega'(q) = \Omega'(q) \cup \{q\}; \qquad \Phi(q) = A \backslash \Omega'(q).$$

Analogout notation is used for subsets of B.

Because $A \backslash \{q\} = \Omega(q) \cup \Phi(q)$, and $S(\vec{0}, \vec{0}) = 1$, we can decompose the excitation as follows: $(N-1)R(A_q, B_x) = \sum_{r \in \Omega(q) \cup \Phi(q)} \sum_{y \in \Omega(x) \cup \Phi(x)} S(\vec{a}_{qr}, \vec{a}_{xy}) C(A_r, B_y) = Z_{11}(q, x) + Z_{12}(q, x) + Z_{21}(q, x) + Z_{22}(q, x)$, with

$$Z_{11}(q, x) = \sum_{r \in \Omega(q)} \sum_{y \in \Omega(x)} (S(\vec{a}_{qr}, \vec{a}_{xy}) C(A_r, B_y));$$

$$Z_{12}(q, x) = \sum_{r \in \Omega(q)} (S(\vec{a}_{qr}, \vec{0}) C(\{r\}, \Phi(x));$$

$$Z_{21}(q, x) = \sum_{b \in \Omega(x)} (S(\vec{0}, \vec{a}_{xy}) C(\Phi(q), \{y\}));$$

$Z_{22}(q, x) = S(\vec{0}, \vec{0}) C(\Phi(a_q), \Phi(b_x)) = C(A, B) - C(\Omega(q), B) - C(\{q\}, B) - C(A, \Omega(x)) + C(\Omega(q), \Omega(x)) + C(\{q\}, \Omega(x)) - C(A, \{x\}) + C(\Omega(a_q), \{x\}) + C(\{q\}, \{x\})$. This decomposition allows us to compute the matrix of $R(a_q, b_x)$ as follows:

1. Each value $Z_{11}(q, x)$ is computable in $O(D^2)$ operations, making the cost for the full set of them $O(N^2D^2)$
2. The full set of row sums $C(\{q\}, B)$ for all q, column sums $C(A, \{x\})$ for all x, and then the total sum $C(A, B)$, can be computed in $O(N^2)$ operations.
3. Since $C(\{q\}, \Phi(x)) = C(\{q\}, B) - C(\{q\}, \Omega'(x))$, each of the values $C(\{q\}, \Phi(x))$ can be obtained, using the pre-computed sums $C(\{q\}, B)$, in $O(D)$ operations, making the cost for the entire matrix of them $O(N^2D)$.
4. Using the pre-computed sums $C(\{q\}, \Phi(x))$, each value $Z_{12}(q, x)$ can be computed in $O(D)$ operations, with the $O(N^2D)$ cost for the entire matrix. Sums $Z_{21}(q, x)$ are computed similarly.
5. Once the row and column sums from step 2 are available, computing each of the nine added values in $Z_{22}(q, x)$ will cost no more than $O(D^2)$, making the cost of the entire matrix $O(N^2D^2)$.

None of the steps above costs more than $O(N^2D^2)$, which means that the total cost of computing the matrix of excitation values $R(A_q, B_x)$ is $O(N^2D^2)$.

The formulas above are significantly simplified in the case of an unweighted unlabeled graph (a conceptual system with a single binary-valued relation type). In such a system $S(\vec{a}_{qr}, \vec{0}) = 0$ for any $r \in \Omega(q)$, and $S(\vec{a}_{qr}, \vec{a}_{xy}) = 1$ for any pair ($r \in \Omega(q)$, $y \in \Omega(x)$). Therefore, $Z_{12}(q, x) = Z_{21}(q, x) = 0$, and $Z_{11}(q, x) = \sum_{r \in \Omega(q)} \sum_{y \in \Omega(x)} C(A_r, B_y)$.

We use a similar transformation to compute the inhibition matrix $I(A_q, B_x)$ in $O(N^2)$ operations at each iteration.

Dynamic Adjustment of Coefficients for Excitation and Inhibition

One problem we observed with the original ABSURDIST algorithm is that, for many systems, especially sparse weighted systems, the correspondence matrix tends to converge to a matrix of all 1s. This is not surprising, considering that in a sparse graph most of the psychological similarities $S(\vec{a}_{qr}, \vec{a}_{xy})$ are equal to $S(\vec{0}; \vec{0}) = 1$, and therefore, most elements of the excitation matrix are close to a positive number, $\sum_a \sum_x C(a, x)$. Thus positive values tend to strongly predominate at each step in the net input matrix

$$N(A_q, B_x) = \alpha E(A_q, B_x) + \beta R(A_q, B_x) - \chi I(A_q, B_x).$$

To prevent the correspondence matrix from becoming saturated, we adjust χ at each step so that the average value of $\alpha E(A_q, B_x) + \beta R(A_q, B_x) - \chi I(A_q, B_x)$ is always zero. This is done as recomputing χ at each step as

$$\chi = \frac{\alpha \sum_q \sum_x E(A_q, B_x) + \beta \sum_q \sum_x R(A_q, B_x)}{\sum_q \sum_x I(A_q, B_x)}. \quad (5)$$

Final mapping selection

To produce a 1-to-1 mapping between the two conceptual systems based on the correspondence matrix $C(a_q, b_x)$, we use the following procedure:

1. Create sets $U_A = \emptyset$, $U_B = \emptyset$. They will be used to store indices of already mapped variables.
2. If U_A covers all concepts from A, or U_B covers all concepts from B, terminate.
3. Select the largest element $C(a_q, b_x)$ in the correspondence matrix such that q does not appear in U_A and x does not appear in U_B. Use random tie-breaking if there are several elements with the same value.
4. Map x to q. Add q to U_A; add x to U_B.
5. Go to step 2.

If the two systems have the same number of concepts ($N_A = N_B = N$), the above algorithm will find a 1-to-1 mapping for all concepts, providing an inexact isomorphism between the two graphs. If $N_A < N_B$, the mapping will be found for all concepts from A, providing an isomorphism between A and an N_A-node subgraph of B. The situation when $N_B < N_A$ is analogous.

Assessing mapping quality

To see how far our mapping P is from an exact isomorphism, we use the *relations mismatch measure* $\mu(A, B, P)$, somewhat similar to the edit distance measure used elsewhere

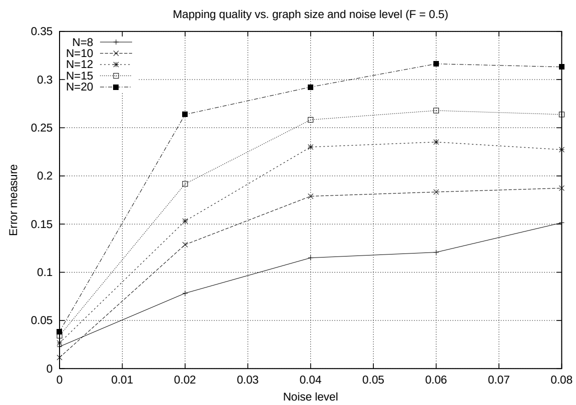

Figure 1: Test 1(a): ε vs. noise level ν and graph size N for fixed graph density $F = 0.5$.

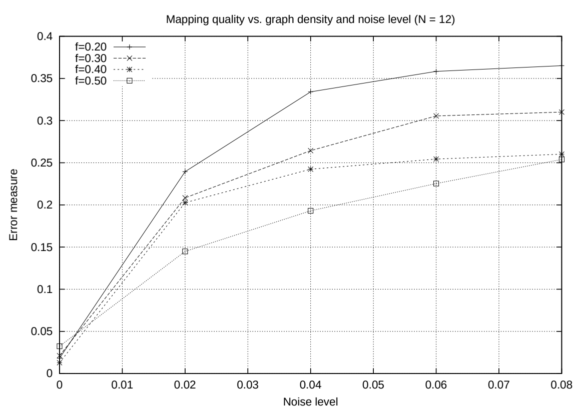

Figure 2: Test 1(b): ε vs. ν and F for fixed $N = 12$.

(Messmer 1995).

$$\mu(A, B, P) = 0.5 \sum_{q=1}^{N} \sum_{r=1}^{N} D_{\mathrm{d}}(a_{qr}, b_{P(q)P(r)}) \quad (6)$$

In the case of an unweighted, unlabeled, undirected graph this measure can be visualized as the "mismapped edge count", i.e. the number of edges that are present in $P(A)$ but absent in the corresponding positions in B, plus the number of edges that are present in $P(B)$ but absent in A.

Performance Evaluation of ABSURDIST II

We have implemented ABSURDIST II in Java. This section describes several experiments carried out on randomly generated graphs. In each experiment $N_R = 100$ runs was performed. In each run, a random conceptual system (the

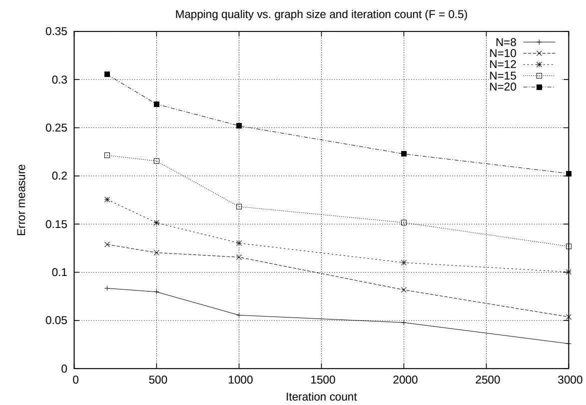

Figure 3: Test 2(a): ε vs. iteration count and graph size for fixed graph density $F = 0.5$ and noise level $\nu = 0.02$.

"original system", S_O) with N nodes and E edges was created. It was matched with ABSURDIST to a "noisy system" S_N generated from S_O by adding noise.

In each experiment, we run our version of ABSURDIST on the translation (S_O, S_N) for 2000-3000 iterations (unless otherwise specified), using $\alpha = 0.0$ (except in Test 4, where $\alpha = 0.5$), and $\beta = (3n - 1)/(2 * n^2 - 2)$. At each step, χ was set as per (5). All elements of the correspondence matrix were initialized with $C(A_q, B_x) = 0.5$.

A good way to measure the quality of mapping P between S_O and S_R found with ABSURDIST II would be to compare it to the best mapping possible. That is, we could use a graph matching algorithm (e.g. (Messmer 1995)) guaranteed to find the best possible isomorphism P_* between the systems with respect to the relation mismatch measure (6), and then look at the adjusted relation mismatch measure $\mu(S_O, S_N, P) - \mu(S_O, S_N, P_*)$. A difference of zero would mean that we have found the best possible mapping.

As finding the guaranteed best isomorphism has exponential complexity, we use a simpler approach to estimate the *base mismatch measure* $\mu(S_O, S_N, P_*)$ in each experiment. First, we assume that, because noise levels are low, the best mapping mapping between S_O and S_R is the identity isomorphism I. Second, we use our knowledge of how the noise was applied to obtain the average *estimated base mismatch measure*, $\mu_* = \langle \mu(S_O, S_N, I) \rangle$, and use μ_* in lieu of the true base mismatch measure $\mu(S_O, S_N, P_*)$.

Finally, we normalize the mismatch measure by dividing it by $2E$, which serves as the estimate for the maximum possible value of the relation mismatch measure between two graphs with E edges each. The resulting *normalized adjusted relation mismatch measure (NARMM)*

$$\varepsilon = \frac{\langle \mu(S_O, S_N, I) \rangle - \mu_*}{2E} \quad (7)$$

is reported for each experiment.

Test 1: Noise tolerance. In this test, S_O was an unlabeled undirected graph with no non-trivial homomorphisms (no

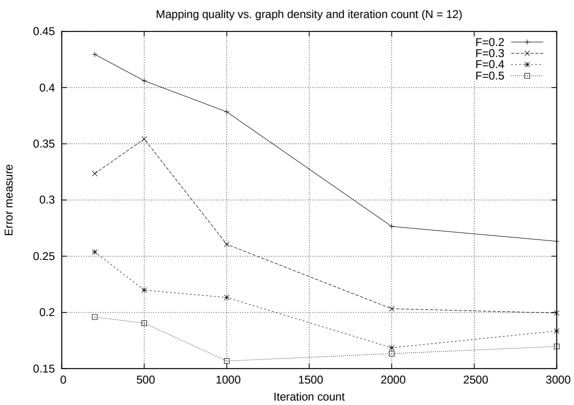

Figure 4: Test 2(b): ε vs. iteration count and graph density for fixed graph size $N = 12$ and noise level $\nu = 0.02$.

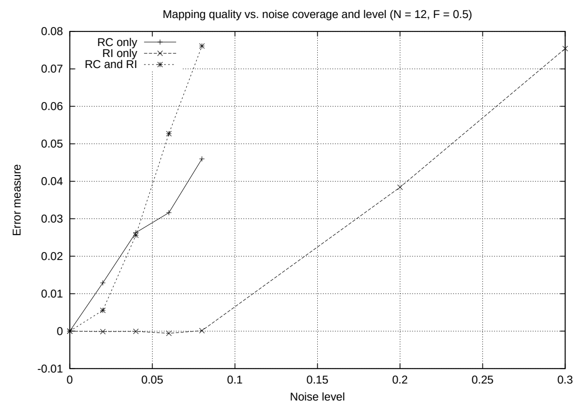

Figure 5: Test 3: ε vs. noise level for coverage noise RC, intensity noise RI, and their combination. $N = 12$, $F = 0.5$.

degrees of symmetry) with N nodes. The number of edges E is chosen as $E = FN(N-1)/2$, where F $(0 < F \leq 0.5)$ is referred to as the *density* of the graph. The noisy system S_N was created as an inexact copy of S_O, where each of $n(n-1)/2$ existing or potential edges was independently switched from present to absent or vice versa with the probability ν. Thus the base mismatch measure could be estimated as $\mu_* = \nu N(N-1)/2$. NARMM computed by (7) with this μ_* after 3000 iterations was reported.

This test shows that, for a fixed number of iterations, fixed learning rate, and $\beta = O(n^{-1})$, the mapping quality deteriorates with increasing graph size (Fig. 1). On the other hand, it improves with increasing graph density (Fig. 2).

Test 2: Iteration count. In this test (Figs. 3,4), we measured how the mapping quality improves with the number of iterations. Pairs of unlabeled, undirected graphs were generated as in Test 1, with noise level $\nu = 0.02$.

Test 2 shows that the rate of convergence to the desired mapping is independent of the graph size. That is, the five lines in Figure 3 are approximately parallel. There is, however, an interaction between graph density and convergence, with sparse graphs converging slower than dense graphs.

Test 3: Coverage noise and intensity noise on weighted graphs. In this test we generated random, undirected weighted graphs with one relation type. The weight of each relation was uniformly distributed on the $[0;1]$ interval.

Two types of noise were tested. In the experiments with *coverage noise* (RC), the difference between the noisy system S_N and the original system S_O consisted in some edges created or destroyed according to the same rules as in Test 1. The creation/destruction probability for each existing or potential edge was ν_C.

In the experiments with the *intensity noise* (RI), S_O and S_N had the same topology, but the intensity of each edge present in S_O was changed in S_N. The difference between the weights of the edges in the two graphs was, for each edge, a random value distributed uniformly on the $[-\nu_C;\nu_C]$ range. However, the intensity change was limited so as to never take the resulting weight outside of the $[0;1]$ interval.

Finally, we carried out experiments with the two types of noise combined. Coverage noise was applied first, and then intensity noise was applied to those edges that existed in S_O and remained in S_N. In the experiments presented here, the two noise levels were equal, $\nu_C = \nu_I$.

The base mismatch measure for a noisy system with any combination of coverage and intensity noise can be estimated as $\mu_* = 0.5 * (\nu_C n(n-1)/2 + (1-\nu_C)\nu_I E)$; this μ_* is in formula (7) to compute NARMM

The results shown in Figure 5 are quite interesting. For a pair of system with intensity noise only, the NARMM is very close to zero within the noise range shown on the graph ($\nu_C \leq 0.08$). This indicates that ABSURDIST II is very tolerant to random variation to the values of weighted edges, matching graphs correctly as long as the topological structure is preserved. This finding may be of importance in applications where the weights of the relations come from real-world data and, by their nature, are imprecise. The NARMM for coverage-noise-only experiments is better than that obtained in similar experiment with an unweighted graph (the $N = 12$ curve in Figure 2, Test 1(b)).

Test 4: External similarity seeding. In this test we explored the effect of external similarity input on ABSURDIST II mapping quality. We used pairs of unlabeled, unweighted graphs designed as in Test 1, with $N = 12$ nodes, density $F = 0.5$, and the noise level ν from 0 to 0.08. Unlike other tests, external similarity was used along with the excitation and inhibition in the net input at each step. The external similarity matrix contained ones in `SNum` randomly chosen diagonal positions, and zeros elsewhere. Thus it provided external "seeding", tying `SNum` concepts in the original system to their counterparts in the noisy system.

The results, shown in Figure 6 for `SNum`$= 0$ through 4 show that the presence of even one external seeding point

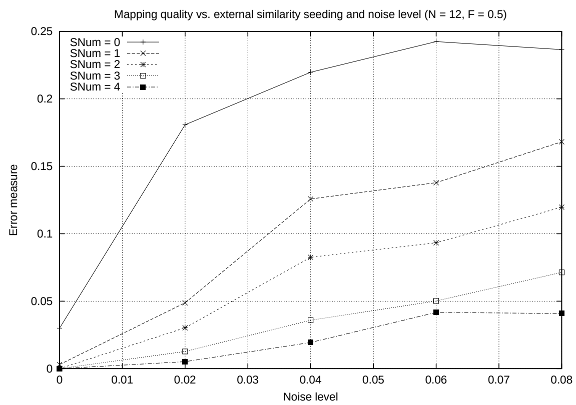

Figure 6: Test 4: ε vs. noise level for various amount of external seeding. `SNum` is the number of non-zeros in the external similarity matrix. $N = 12$, $F = 0.5$.

improves mapping quality significantly compared to the results obtained with internal similarity only. Adding extra seeding points improves the mapping quality further, but the marginal gains diminish as more correspondences are externally seeded. The improvements in mapping garnered by seeding correspondences far outstrips the gains predicted if only the seeded correspondence itself was correctly mapped. This is due to the ripples of influence that one seeded correspondence has on fixing other correspondences.

Conclusion and Future Work

We introduced the attributed graph representation of conceptual systems, extending the ABSURDIST framework to handle systems with multiple relation types, including weighted and directed relations. We exploited sparsity of the relation graphs to improve efficiency and scalability of the algorithm.

We illustrated the behavior of ABSURDIST II on some simple models. We have shown that our iterative approach inspired by constraint-propagation neural networks can be used to match concept systems represented by purely topological structures (unweighted graphs). When noise is introduced in the form of adding and deleting edges, the resulting translation is considerably poorer than when noise is added to the weights. That is, distorting a system by "flipping" X% of the edges is more damaging than distorting a system by altering weights all of the edges by X%.

Future work may include applying the algorithm to conceptual systems from real-world domains: ontologies, dictionaries, database schemas. We have already achieved some success using ABSURDIST II for matching hundreds of terms in bilingual texts. Further investigation of the convergence properties of the ABSURDIST II is worthwhile, especially if it can lead to redesigning the process to guarantee convergence. It would be useful to compare the effectiveness and efficiency of our approach to those of already implemented classical methods, such as (Messmer & Bunke 1996), guaranteed to find the best mapping with respect to the target mismatch function.

Still, in terms of real-world translation tasks, ABSURDIST II already offers benefits over many existing graph matching algorithms: it does not enforce 1-to-1 mappings and so allows multiple concepts from a richer conceptual system to map onto a single concept from a more impoverished system, it provides a natural way to incorporate both internal and external sources of information, and it accommodates several different graph structures. More generally, it offers the promise of showing how the meaning of a concept can be based simultaneously on its role within its conceptual system and on its external grounding, and that these two sources of information cooperate rather than compete.

Acknowledgments

This research was supported by a grant from the Lockheed Martin Corporation. We would like to thank Todd Hughes, Brian Rogosky, and Ben Ashpole for helpful comments.

References

Gold, S., and Rangarajan, A. 1996. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(4):377–388.

Goldstone, R. L., and Rogosky, B. J. 2002. Using relations within conceptual systems to translate across conceptual systems. *Cognition* 295–320.

Lenat, D., and Feigenbaum, E. 1991. On the thresholds of knowledge. *Artificial Intelligence* 47:185–250.

Melnik, S.; Garcia-Molina, H.; and Rahm, E. 2002. Similarity flooding: a versatile graph matching algorithm and its application to schema matching. In *18th International Conference on Data Engineering (ICDE)*, 117–128.

Messmer, B., and Bunke, H. 1996. Fast error-correcting graph isomorphism based on model precompilation. Technical report, Institut fur Informatik, Universität Bern.

Messmer, B. T. 1995. *Efficient Graph Matching Algorithms*. Ph.D. Dissertation, Universität Bern.

Pelillo, M. 1998. Replicator equations, maximal cliques, and graph isomorphism.

Rangarajan, A., and Mjolsness, E. 1994. A Lagrangian relaxation network for graph matching. In *Proc. Int. Conf. Neural Networks*, volume 7, 4629–4634. Inst. Electrical & Electronics Engineers.

Schadler, K., and Wysotzki, F. 1997. A connectionist approach to structural simiarity determination as a basis of clustering, classification and feature detection. In *Principles of Data Mining and Knowledge Discovery*, 254–264.

Schädler, K., and Wysotzki, F. 1999. Comparing structures using a Hopfield-style neural network.