Mapping WordNets Using Structural Information

J. Daudé, L. Padró & G. Rigau
TALP Research Center
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya. Barcelona
{daude,padro,g.rigau}@lsi.upc.es

Abstract

We present a robust approach for linking already existing lexical/semantic hierarchies. We used a constraint satisfaction algorithm (relaxation labeling) to select – among a set of candidates– the node in a target taxonomy that bests matches each node in a source taxonomy. In particular, we use it to map the nominal part of WordNet 1.5 onto WordNet 1.6, with a very high precision and a very low remaining ambiguity.

1 Introduction

There is an increasing need of having available general, accurate and broad coverage multilingual lexical/semantic resources for developing NL applications. Thus, a very active field inside NL during the last years has been the fast development of generic language resources.

Several attempts have been performed to connect already existing ontologies. In (Ageno et al., 1994), a Spanish/English bilingual dictionary is used to (semi)automatically link Spanish and English taxonomies extracted from DGILE (Alvar, 1987) and LDOCE (Procter, 1987). Similarly, a simple automatic approach for linking Spanish taxonomies extracted from DGILE to WordNet (Miller et al., 1991) synsets is proposed in (Rigau et al., 1995). The work reported in (Knight and Luk, 1994) focuses on the construction of Sensus, a large knowledge base for supporting the Pangloss machine translation system. In (Okumura and Hovy, 1994) (semi)automatic methods for associating a Japanese lexicon to an English ontology using a bilingual dictionary are described. Several experiments aligning EDR and WordNet ontologies are described in (Utiyama and Hasida, 1997). Several lexical resources and techniques are combined in (Atserias et al., 1997) to map Spanish words from a bilingual dictionary to WordNet, and in (Farreres et al., 1998) the use of the taxonomic structure derived from a monolingual MRD is proposed as an aid to this mapping process.

The use of relaxation labeling algorithm to attach substantial fragments of the Spanish taxonomy derived from DGILE (Rigau et al., 1998) to the English WordNet using a bilingual dictionary for connecting both hierarchies, has been reported in (Daudé et al., 1999).

In this paper we use the same technique to map WN1.5 to WN1.6. The aim of the experiment is twofold: First, show that the method is general enough to link any pair of ontologies. Second, evaluate our taxonomy linking procedure, by comparing our results with other WN1.5 to WN1.6 existing mappings.

This paper is organized as follows: In section 2 we describe the used technique (the relaxation labeling algorithm) and its application to hierarchy mapping. In section 3 we describe the constraints used in the relaxation process, and finally, after presenting some experiments and results, we offer some conclusions and outline further lines of research.

2 Application of Relaxation Labeling to NLP

Relaxation labeling (RL) is a generic name for a family of iterative algorithms which perform function optimization, based on local information. See (Torras, 1989) for a summary. Its most remarkable feature is that it can deal with any kind of constraints, thus, the model can be improved by adding any constraints available, and the algorithm is independent of the complexity of the model. That is, we can use more sophisticated constraints without changing the algorithm.

The algorithm has been applied to POS tagging (Màrquez and Padró, 1997), shallow parsing (Voutilainen and Padró, 1997) and to word sense disambiguation (Padró, 1998).

Although other function optimization algorithms could have been used (e.g. genetic algorithms, simulated annealing, etc.), we found RL to be suitable to our purposes, given its ability to use models based on context constraints, and the existence of previous work on applying it to NLP tasks.

Detailed explanation of the algorithm can be found in (Torras, 1989), while its application to NLP tasks, advantages and drawbacks are addressed in (Padró, 1998).

2.1 Algorithm Description

The Relaxation Labeling algorithm deals with a set of variables (which may represent words, synsets, etc.), each of which may take one among several different labels (POS tags, senses, MRD entries, etc.). There is also a set of constraints which state compatibility or incompatibility of a combination of pairs variable–label.

The aim of the algorithm is to find a weight assignment for each possible label for each variable, such that (a) the weights for the labels of the same variable add up to one, and (b) the weight assignment satisfies –to the maximum possible extent– the set of constraints.

Summarizing, the algorithm performs constraint satisfaction to solve a consistent labeling problem. The followed steps are:

1. Start with a random weight assignment.
2. Compute the *support* value for each label of each variable. Support is computed according to the constraint set and to the current weights for labels belonging to context variables.
3. Increase the weights of the labels more compatible with the context (larger support) and decrease those of the less compatible labels (smaller support). Weights are changed proportionally to the support received from the context.
4. If a stopping/convergence criterion is satisfied, stop, otherwise go to step 2. We use the criterion of stopping when there are no more changes, although more sophisticated heuristic procedures may also be used to stop relaxation processes (Eklundh and Rosenfeld, 1978; Richards et al., 1981).

The cost of the algorithm is proportional to the product of the number of variables by the number of constraints.

2.2 Application to taxonomy mapping

As described in previous sections, the problem we are dealing with is to map two taxonomies. In this particular case, we are interested in mapping WN1.5 to WN1.6, that is, assign each synset of the former to at least one synset of the later.

The modeling of the problem is the following:

- Each WN1.5 synset is a variable for the relaxation algorithm. We will refer to it as *source synset* and to WN1.5 as *source taxonomy*.
- The possible labels for that variable are all the WN1.6 synsets which contain a word belonging to the source synset. We will refer to them as *target synsets* and to WN1.6 as *target taxonomy*.
- The algorithm will need constraints stating whether a WN1.6 synset is a suitable

assignment for a WN1.5 synset. As described in section 3, these constraints will rely on the taxonomy structure.

3 The Constraints

Constraints are used by the relaxation labeling algorithm to increase or decrease the weight for a variable label. In our case, constraints increase the weights for the connections between a source synset and a target synset. Increasing the weight for a connection implies decreasing the weights for all the other possible connections for the same source synset. To increase the weight for a connection, constraints look for already connected nodes that have the same relationships in both taxonomies.

Although there is a wide range of relationships between WordNet synsets which can be used to build constraints, we have focused on the hyper/hyponym relationships. That is, we increase the weight for a connection when the involved nodes have hypernyms/hyponyms also connected. We consider hyper/hyponym relationships either directly or indirectly (i.e. ancestors or descendants), depending on the kind of constraint used.

Figure 1 shows an example of possible connections between two taxonomies. Connection C_4 will have its weight increased due to C_5, C_6 and C_1, while connections C_2 and C_3 will have their weights decreased.

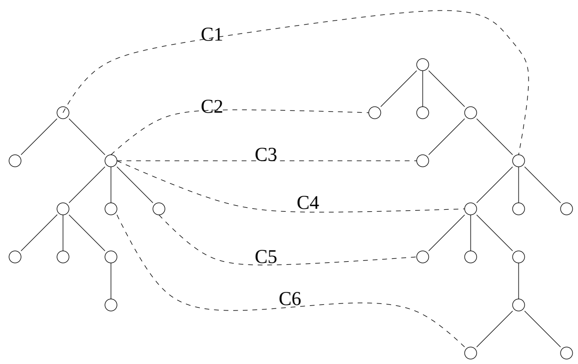

Figure 1: Example of connections between taxonomies.

We distinguish different kinds of constraints, depending on whether we consider hyponyms, hypernyms or both, on whether we consider those relationships direct or indirect, and on in which of both taxonomies we do so. Each constraint may be used alone or combined with others.

Below we describe all kinds of constraint used. They are labeled with a three–character code (XYZ), which must be read as follows: The first character (X) indicates how the hyper/hyponym relationship is considered in the source taxonomy: only for *immediate* nodes (I) or for *any* (A) ancestor/descendant. The second character (Y) codes the same information for the target taxonomy side. The third character indicates whether the constraints requires the existence of a connected hypernym (E), hyponym (O), or both (B).

IIE **constraint:** The simplest constraint is to check whether the connected nodes have respective direct hypernyms also connected. IIE stands for immediate source(I), immediate target (I) hypernym (E).

This constraint will increase the weights for those connections in which the immediate hypernym of the source node is connected with the immediate hypernym of the target node.

IIO **constraint:** This constraint increases the weight for that connections in which an immediate hyponym of the source node is connected to an immediate hyponym of the target node.

IIB **constraint:** This constraint increases the weight for the connections in which the immediate hypernym of the source node is connected to the immediate hypernym of the target node and an immediate hyponym of the source is connected to an immediate hyponym of the target.

II **constraints.** If we use constraints IIE, IIO and IIB at the same time, weights will be modified for words matching any of the constraints. That is, we are additively combining constraints. In the case where two of them apply, their effects will be added. If they have opposite effects, they will cancel each other. Figure 2 shows a graphical representation of all II constraints.

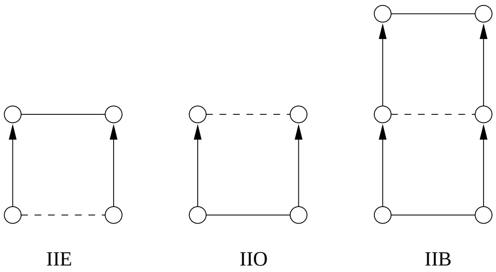

Figure 2: II constraints.

The arrows indicate an immediate hypernymy relationship. The nodes on the left hand side correspond to the source taxonomy and the nodes on the right to the target hierarchy. The dotted line is the connection which weight will be increased due to the existence of the connection indicated with a continuous line.

AIE **constraint:** This constraint increases the weight for the connections in which an ancestor of the source node is connected to the immediate hypernym of the target node.

AIO **constraint:** This constraint increases the weight for the connections in which a descendant of the source node is connected to an immediate hyponym of the target node.

AIB **constraint:** This constraint increases the weight for the connections in which an ancestor of the source node is connected to the immediate hypernym of the target node and a descendant of the source node is connected to an immediate hyponym of the target node.

AI **constraints.** If we use constraints AIE, AIO and AIB simultaneously, we apply either a hypernym constraint, either a hyponym constraint or either both of them. In the last case, the joint constraint is also applied. This means than connections with matching hypernym and hyponym will have their weights doubly increased. Figure 3 shows a graphical representation of all AI constraints.

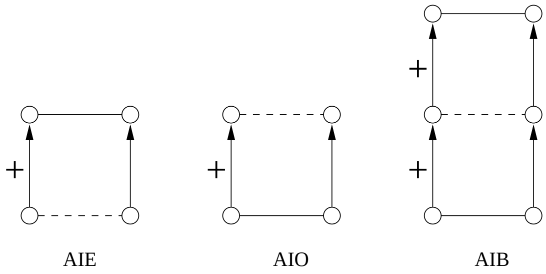

Figure 3: AI constraints.

In this figure, the + sign indicates that the hypernymy relationship represented by the arrow does not need to be immediate. In this case, this iteration is only allowed in the source taxonomy.

IA **constraints:** Are symmetrical to AI constraints. In this case, recursion is allowed only on the target taxonomy.

Figure 4 shows a graphical representation of all IA constraints.

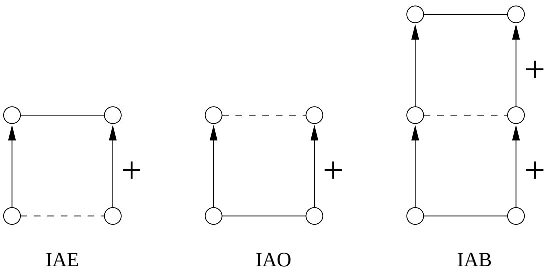

Figure 4: IA constraints.

AA **constraints:** Include the above combinations, but allowing recursion on both sides.

Figure 5 shows a graphical representation of all AA constraints.

4 Experiments and Results

In the performed tests we used simultaneously all constraints with the same recursion pattern. This yields the packs: II, AI, IA and AA.

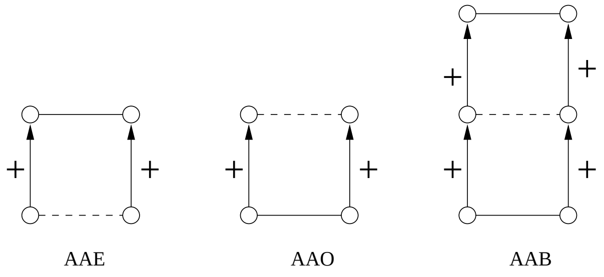

Figure 5: AA constraints.

Results are reported only for the later, since it is the most informed constraint set.

We also compared our mapping with the *SenseMap* provided by Princeton[1], and the coincidence was quite high, specially in the cases in which *SenseMap* has a high confidence score. Details can be found in section 4.1.

In order to perform the comparison, we had to convert *SenseMap*, which is a sense mapping (that is, it maps each *variant* in WN1.5 to a variant in WN1.6), into a *synset* mapping, which is what our algorithm does. Since synsets are coarser than senses, the conversion is straightforward. When two senses in the same 1.5 synset were assigned two senses in different 1.6 synsets, we took both targets as valid, slightly increasing the remaining ambiguity of *SenseMap*.

The results are computed over the synsets with at least one candidate connection, which represent 99.1% of WN1.5. We consider *ambiguous* synsets those with more than one candidate connection.

Table 1 presents the amount of nodes for which disambiguation is performed, and some candidate connections discarded (i.e. they do not keep as possible *all* the candidates).

	ambiguous	overall
SenseMap	98.0%	99.2%
RL	99.8%	99.9%

Table 1: Coverage of WN1.5 for both mappings.

Table 2 presents an estimation of how many of those assignment were right, as well as the precision for *SenseMap*, computed under the same conditions. Those figures were computed by manually linking to WN1.6 a sample of 1900 synsets randomly chosen from WN1.5, and then use this sample mapping as a reference to evaluate all mappings. These figures show that our system performs a better mapping than *SenseMap*. The difference between both mappings is significant at a 95% confidence level.

	ambiguous	overall
SenseMap	93.3%–96.9%	96.9%–98.6%
RL ($\delta = 0.3$)	96.5%–97.7%	98.4%–98.9%
RL ($\delta = 0.4$)	97.0%–97.6%	98.6%–98.9%
RL ($\delta = 0.5$)	97.2%–97.6%	98.7%–98.9%

Table 2: Precision–recall results for both WN1.5–WN1.6 mappings.

Since relaxation labeling performs a weight assignment for each possible connection, we can control the remaining ambiguity (and thus the recall/precision tradeoff) by changing the threshold (δ) that the weight for a connection has to reach to be considered a solution. Although higher thresholds maintain recall and produce a higher precision, differences are not statistically significant.

4.1 Coincidence of Both Mappings

For each confidence group in the Princeton mapping, the *soft agreement* column in table 3 indicates the percentage of WN1.5 synsets in which our system proposes at least one connection also proposed by the Princeton mapping. The *hard agreement* column indicates the amount of connections proposed by our system also proposed by Princeton mapping.

The agreement between both systems is quite high, specially for the groups with a high confidence level. This is quite reasonable, since a perfect system would be expected to agree with the assignments in 20% confidence group of *SenseMap* only about 20% of the times. It also must be taken into account that for low confidence groups, *SenseMap* is much more ambiguous.

[1]See WN web page at http://www.cogsci.princeton.edu/~wn/

confidence group	Agreement $\delta = 0.3$ hard	soft	$\delta = 0.4$ hard	soft	$\delta = 0.5$ hard	soft
monosemous	96.9%	97.3%	97.0%	97.3%	97.1%	97.2%
100%	88.6%	90.4%	89.1%	90.1%	89.5%	89.8%
90%	87.9%	89.8%	88.4%	89.3%	88.7%	89.1%
80%	69.3%	70.2%	70.1%	70.5%	70.4%	70.4%
70%	76.5%	78.0%	76.4%	77.6%	76.5%	76.8%
60%	53.8%	53.8%	53.8%	53.8%	53.8%	53.8%
50%	68.4%	81.2%	72.7%	77.4%	72.7%	77.4%
40%	50.7%	50.8%	50.8%	50.8%	50.8%	50.8%
30%	65.3%	65.3%	65.3%	65.3%	65.3%	65.3%
20%	32.6%	32.6%	32.6%	32.6%	32.6%	32.6%
subtotal	87.3%	89.1%	87.8%	88.8%	88.3%	88.6%
Total	93.6%	94.5%	93.8%	94.4%	94.1%	94.2%

Table 3: Agreement between both mappings.

The average remaining ambiguity in Princeton mapping and in the mapping performed by the relaxation labeling algorithm is shown respectively in columns *SenseMap ambiguity* and RL *ambiguity* of table 4.

Our system proposes, in most cases, a unique WN1.6 synset for each WN1.5 synset. The average ranges from 1.001 to 1.007 proposals per synset depending on the chosen δ threshold, while the Princeton mapping has an average of 1.007.

Summarizing, the obtained results point that our system is able to produce a less ambiguous assignment than *SenseMap*, with a significantly higher accuracy and wider coverage.

In addition, our system only uses structural information (namely, hyper/hyponymy relationships) while *SenseMap* uses synset words, glosses, and other information in WordNet. On the one hand, when information other than taxonomy structure is used results might be even better. On the other hand, for cases in which such information is not available (e.g. further development of EuroWordNets in new languages), structure may provide a reliable basis.

5 Conclusions & Further Work

We have applied the relaxation labeling algorithm to assign an appropriate node in a target taxonomy to each node in a source taxonomy, using only hyper/hyponymy information.

Results on WN1.5 to WN1.6 mapping have been reported. The high precision achieved provides further evidence that this technique –previously used in (Daudé et al., 1999) to link a Spanish taxonomy to WN1.5– constitutes an accurate method to connect taxonomies, either for the same or different languages. Further extensions of this technique to include information other than structural may result in a valuable tool for those concerned with the development and improvement of large lexical or semantic ontologies.

The results obtained up to now seem to indicate that:

- The relaxation labeling algorithm is a good technique to link two different hierarchies. For each node with several possible connections, the candidate that best matches the surrounding structure is selected.

- The structural information provides enough knowledge to accurately link taxonomies. Experiments on mapping tax-

Confidence group	Size	*SenseMap* ambiguity	RL ambiguity $\delta = 0.3$	$\delta = 0.4$	$\delta = 0.5$
monosemous	45807	1.003	1.001	1.001	1.001
100%	20075	1.000	1.020	1.011	1.003
90%	2977	1.007	1.022	1.010	1.005
80%	326	1.080	1.018	1.009	1.000
70%	249	1.024	1.020	1.012	1.004
60%	93	1.043	1.000	1.000	1.000
50%	32	1.063	1.125	1.064	1.064
40%	67	1.448	1.031	1.015	1.000
30%	65	1.569	1.000	1.000	1.000
20%	209	2.215	1.031	1.025	1.020
subtotal	24093	1.016	1.020	1.011	1.003
Total	69900	1.007	1.007	1.006	1.001

Table 4: Average remaining ambiguity of both mappings.

onomies automatically extracted from a Spanish MRD to WN1.5 (Daudé et al., 1999) show that the technique may be useful even when both taxonomies belong to different languages or have structures less similar than in the case reported in this paper.

- The system produces a good assignment for WN mapping, based only on hyper/hyponymy relationships, which is specially useful when no other information is available (i.e. in the case of mapping the EuroWN hierarchies). The remaining ambiguity is low with a high accuracy, and precision–recall tradeoff may be controlled by adjusting the δ threshold.

Some issues to address for improving the algorithm performance, and to exploit its possibilities are:

- Use other relationships than hyper/hyponymy as constraints to select the best connection. Relationships as sibling, cousin, etc. could be used. In addition, WN provides other relationships such as synonymy, meronymy, etc. which could also provide useful constraints.
- Use other available information, such as synset words, glosses, etc. in the WN to WN mapping task.
- Link the verbal, adjectival, and adverbial parts of WN1.5 and WN1.6.
- Test the performance of the technique to link other structures (e.g WN-EDR, WN-LDOCE, dutch-WN, italian-WN, ...).
- Use it to link taxonomies for new languages to EuroWordNet.
- Give a step beyond the source-to-target vision, and map the taxonomies in a symmetrical philosophy, that is, each node of each taxonomy is assigned to a node in the other taxonomy. This should increase the coverage, and reinforce/discard connections that may be weak when assigned only in one direction. This could even open the doors to many-to-many taxonomy mapping.

6 Acknowledgments

This research has been partially funded by the the UE Commission (NAMIC IST-1999-12392), by the Spanish Research Department (TIC98-423-C06-06) and by the Catalan Research Department, through the CREL project and the Quality Research Group Programme (GRQ93-3015).

References

A. Ageno, I. Castellón, F. Ribas, G. Rigau, H. Rodríguez, and A. Samiotou. 1994. TGE: Tlink Generation Environment. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING'94)*, Kyoto, Japan.

M. Alvar, editor. 1987. *Diccionario General Ilustrado de la Lengua Española VOX*. Biblograf S.A, Barcelona, Spain.

J. Atserias, S. Climent, X. Farreres, G. Rigau, and H. Rodríguez. 1997. Combining Multiple Methods for the Automatic Construction of Multilingual WordNets. In *proceedings of International Conference on Recent Advances in Natural Language Processing (RANLP'97)*, Tzigov Chark, Bulgaria.

J. Daudé, L. Padró, and G. Rigau. 1999. Mapping Multilingual Hierarchies Using Relaxation Labeling. In *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC'99).*, Maryland, US.

J. O. Eklundh and A. Rosenfeld. 1978. Convergence Properties of Relaxation Labelling. Technical Report 701, Computer Science Center. University of Maryland.

X. Farreres, G. Rigau, and H. Rodríguez. 1998. Using WordNet for Building WordNets. In *Proceedings of COLING-ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, Montréal, Canada.

K. Knight and S. Luk. 1994. Building a Large-Scale Knowledge Base for Machine Translation. In *Proceedings of the American Association for Artificial Inteligence (AAAI'94).*

L. Màrquez and L. Padró. 1997. A Flexible POS Tagger Using an Automatically Acquired Language Model. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics. Joint ACL/EACL*, pages 238–245, Madrid, Spain, July.

G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. 1991. Five Papers on WordNet. *International Journal of Lexicography.*

A. Okumura and E. Hovy. 1994. Building japanese-english dictionary based on ontology for machine translation. In *proceedings of ARPA Workshop on Human Language Technology*, pages 236–241.

L. Padró. 1998. *A Hybrid Environment for Syntax–Semantic Tagging.* Phd. Thesis, Dep. Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya, February. http://www.lsi.upc.es/~padro.

P. Procter, editor. 1987. *Longman Dictionary of Common English.* Longman Group, Harlow, Essex, England.

J. Richards, D. Landgrebe, and P. Swain. 1981. On the accuracy of pixel relaxation labelling. *IEEE Transactions on Systems, Man and Cybernetics*, 11(4):303–309.

G. Rigau, H. Rodríguez, and J. Turmo. 1995. Automatically extracting Translation Links using a wide coverage semantic taxonomy. In *proceedings 15th International Conference AI'95*, Montpellier, France.

G. Rigau, H. Rodríguez, and E. Agirre. 1998. Building Accurate Semantic Taxonomies from MRDs. In *Proceedings of COLING-ACL'98*, Montréal, Canada.

C. Torras. 1989. Relaxation and Neural Learning: Points of Convergence and Divergence. *Journal of Parallel and Distributed Computing*, 6:217–244.

M. Utiyama and K. Hasida. 1997. Bottom-up Alignment of Ontologies. In *Proceedings of IJCAI workshop on Ontologies and Multilingual NLP*, Nagoya, Japan.

A. Voutilainen and L. Padró. 1997. Developing a Hybrid NP Parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing, ANLP*, pages 80–87, Washington DC. ACL.