Experiments on Applying Relaxation Labeling to Map Multilingual Hierarchies.

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

Abstract

This paper explores the automatic construction of a multilingual Lexical Knowledge Base from preexisting lexical resources. This paper presents a new approach for linking already existing hierarchies. The Relaxation labeling algorithm is used to select –among all the candidate connections proposed by a bilingual dictionary– the right conection for each node in the taxonomy.

1 Introduction

There is no doubt about the increasing need of owning accurate and broad coverage general lexical/semantic resources for developing NL applications. Thus, one of the main issues in last years as regards NLP activities has been focused on the fast development of generic language resources. These resources include lexicons, lexical databases (LDBs), lexical knowledge bases (LKBs), ontologies, etc.

Special interest presents, for knowledge-based NLP tasks, the availability of wide coverage ontologies. Most known ontologies (suchas GUM, CYC, ONTOS, MICROKOSMOS, EDR or WordNet, see [Gó98] for an extensive survey) differ in great extent on several characteristics (e.g. broad coverage vs. domain specific, lexically oriented vs. conceptually oriented, granularity, kind of information stored, kind of relations, way of being built, etc.). However, the success of WordNet has determined the emergence of several projects that aim the construction of WordNets for other languages than English (e.g., [HF97, AMS97]) or to develop multilingual WordNets The most important project in this line is EuroWordNet (EWN) [PBR+97].

The construction of a WN for a specific language L_g (L_gWN) can be tackled in different ways, depending on the lexical sources available. Of course the manual construction can be undertaken quite straightforwardly and leads to the best results in terms of accuracy [BCE+98], But this methodology has the important drawback of its cost. So, other approaches have been carried out taking profit of available resources in fully automatic or semi-automatic ways. Basically four kinds of resources have been used:

1. English WN (EnWN), as an initial skeleton for trying to attach the words from Lg language to it.
2. Already existing taxonomies of L_g (both at word and at sense level),
3. Bilingual (English and L_g) and
4. Monolingual L_g dictionaries.

All the approaches using EnWN as skeleton are based on the assumption of a close conceptual similarity between English and L_g, in such a way that most of the structure (relations) in EnWN could be maintained for L_gWN.

In the case of bilingual dictionaries the usual approach is to try to link the English counterpart of entries to synsets in EnWN and to assume that the entry can be linked to the same synset [ACF+97]. Monolingual dictionaries have been used basically as a source for extracting taxonomic (hypernym) links between words (or senses [BG92], [RRA98]) and in lower extent for extracting other kinds of semantic relations [Ric97] (e.g. meronymic links).

Once a taxonomy of L_g (already existing or built from a monolingual MRD) is available, the task can consist of 1) enriching the taxonomic structure with other semantic links (manually or automatically), as is the case of building individual WNs, or 2) merging this structure with other already existing ontologies (as EnWN or EWN).

Recently, several attempts have been performed to produce multilingual ontologies. [ACR+94] use a Spanish/English bilingual dictionary for (semi)automatically linking Spanish and English taxonomies extracted from DGILE and LDOCE. In a similar approach, [RRT95] propose an automatic approach for linking Spanish taxonomies extracted from DGILE to WordNet synsets. [KL94] focus on the construction of Sensus, a large knowledge base for supporting the Pangloss machine traslation system merging ontologies (ONTOS and Upper Model) and WordNet with monolingual and bilingual MRDs. [OH94] describe (semi)automatic methods for associating a Japanese lexicon to an English ontology using a bilingual dictionary. [UH97] describe several experiments aligning EDR and WordNet ontologies. [ACF+97] combine several lexical resources and techniques to map Spanish words from a bilingual dictionary to WordNet in order to build a parallel in structure semantic net. [FRR98] propose also the use of the taxonomic structure derived form a monolingual MRD to aid this mapping process.

This paper presents a novel approach for merging already existing hierarchies. The method has been applied to attach substantial fragments of the Spanish taxonomy derived from DGILE (see [RRA98]) to the English WordNet using a bilingual dictionary for connecting both hierarchies. Although we used the system for connecting two hierarchies of different languages, we are especting better results when appliying the method on different versions of the same hierarchy (for instance, WN1.5 and WN1.6). Thus, using this methodology, we are able to reuse lexical resources developed previously for older versions of the same hierarchy (as for instance, SemCor).

2 Application of Relaxation Labelling to NLP

This section discusses the the relaxation labelling algorithm and its use to perform NLP tasks. To enable the application of relaxation labelling, the language model must be described in terms of algorithm elements –variables, labels, constraints, etc.–.

Although relaxation labelling has been mainly used in fields other than NLP (engineering, computer vision, ...), some researchers in optimization techniques [PR94, PM94] have used POS tagging as a toy problem to experiment their methods to improve the performance of relaxation labelling. They used a 1000-word test corpus, and only binary constraints, which was enough to their purposes of testing a method for estimating constraint compatibility values. From the NLP field, the works by [Pad96, MP97, VP97, Pad98] constitute a reliable background to state that relaxation labelling may be a useful tool for the NLP interests.

We will describe the relaxation labelling algorithm from a general point of view in section 2.1. Afterwards, the way to use it to match taxonomies will be described.

2.1 Algorithm Description

In this section the relaxation algorithm is described from a general point of view.

Let $V = \{v_1, v_2, \ldots, v_N\}$ be a set of variables.

Let $T_i = \{t_1^i, t_2^i, \ldots, t_{m_i}^i\}$ be the set of possible labels for variable v_i (where m_i is the number of different labels that are possible for v_i).

Let C be a set of constraints between the labels of the variables. Each constraint is a "compatibility value" for a combination of pairs variable–label. For instance, the constraint

$$0.53 \quad [(v_1, A)(v_3, B)]$$

states that the combination of variable v_1 having label A, and variable v_3 having label B has a compatibility value of 0.53. Constraints can be of any order, so we can define the compatibility value for combinations of any number of variables (obviously we can have combinations of at most N variables).

The aim of the algorithm is to find a *weighted labelling* such that *global consistency* is maximized.

A *weighted labelling* is a weight assignation for each possible label of each variable:

> $P = (p^1, p^2, \ldots, p^N)$ where each p^i is a vector containing a weight for each possible label of v_i, that is: $p^i = (p_1^i, p_2^i, \ldots, p_{m_i}^i)$

Since relaxation is an iterative process, when the time step is relevant, we will note the weight for label j of variable i at time n as $p_j^i(n)$. When the time step is not relevant, we will note it as p_j^i.

Maximizing *global consistency* is defined as maximizing for each variable v_i, $(1 \leq i \leq N)$, the average support for that variable, which is defined as the weighted sum

of the support received by each of its possible labels, that is:

$$\sum_{j=1}^{m_i} p_j^i \times S_{ij}$$

where p_j^i is the weight for label j of variable v_i and S_{ij} is the support received by that pair from the context. The support for the pair variable–label expresses *how compatible* that pair is with the labels of neighbouring variables, according to the constraint set.

The performed *global consistency* maximization is a vector optimization. It does not maximize –as one might think– the sum of the supports of all variables. It finds a weighted labelling such that any other choice would not increase the support for *any* variable given –of course– that such a labelling exists. If such a labelling does not exist, the algorithm will end in a local maximum.

The relaxation algorithm consists of:

- start in a random labelling P_0.
- for each variable, compute the "support" that each label receives from the current weights for the labels of the other variables (i.e. see how compatible is the current weighting with the current weightings of the other variables, given the set of constraints).
- Update the weight of each variable label according to the support obtained by each of them (that is, increase weight for labels with high support, and decrease weight for those with low support).
- iterate the process until a convergence criterion is met.

The support computing and weight changing must be performed in parallel, to avoid that changing a weight for a label would affect the support computation of the others.

We could summarize this algorithm saying that at each time step, a variable changes its label weights depending on how compatible is that label with the labels of the other variables at that time step. If the constraints are consistent, this process converges to a state where each variable has weight 1 for one of its labels and weight 0 for all the others.

Note that the *global consistency* idea –defined as the maximization of the average support received by each variable from the context– makes the algorithm robust, since the problem of having mutually incompatible constraints (so one can not find a combination of label assignations which satisfies all the constraints) is solved because relaxation does not (necessarily) find an exclusive combination of labels, that is, an unique label for each variable, but a weight for each possible label such that consistency is maximized (the constraints are satisfied to the maximum possible degree).

Advantages of the algorithm are:

- Its highly local character (each variable can compute its new label weights given only the state at previous time step). This makes the algorithm highly parallelizable (we could have a processor to compute the new label weights for each variable, or even a processor to compute the weight for each label of each variable).
- Its expressiveness, since we state the problem in terms of constraints between variable labels.
- Its flexibility, we do not have to check absolute consistency of constraints.
- Its robustness, since it can give an answer to problems without an exact solution (incompatible constraints, insufficient data, ...)
- Its ability to find locally optimal solutions to NP problems in a non-exponential time (Only if we have an upper bound for the number of iterations, i.e. convergence is fast or the algorithm is stopped after a fixed number of iterations).

Drawbacks of the algorithm are:

- Its cost. Being N the number of variables, v the average number of possible labels per variable, c the average number of constraints per label, and I the average number of iterations until convergence, the average cost is $N \times v \times c \times I$, that is, it depends linearly on N, but for a problem with many labels and constraints, or if convergence is not quickly achieved, the multiplying terms might be much bigger than N.
- Since it acts as an approximation of gradient step algorithms, it has their typical convergence problems: Found optima are local, and convergence is not guaranteed, since the chosen step might be too large for the function to optimize.
- In general, constraints must be written manually, since they are the modelling of the problem. This is good for easy-to-model domains or reduced constraint-set problems, but in the case of POS tagging or WSD constraint are too many and too complicated to be written by hand.
- The difficulty to state which is the *compatibility value* for each constraint. If we deal with combinatorial problems with an exact solution (e.g. travelling salesman), the constraints will be all fully compatible (e.g. stating that it is possible to go to any city from any other) or fully incompatible (e.g. stating that it is not possible to be twice in the same city). But if we try to model more sophisticated or less exact problems (such as POS tagging) things will not be black or white. We will have to assign a compatibility value to each constraint.
- The difficulty to choose the support and updating functions more suitable for each particular problem.

2.1.1 Support Function

The relaxation labelling algorithm requires a way to compute which is the support for a variable label given the constraints and the current label weights for the other

variables. This is called the *support function* and it is the heart of the algorithm, since it is closely related to what will be maximized.

To define the support received by a variable label from its context, we have to combine the individual influences of each constraint that can be applied for that pair in the current context. So, we will define $Inf(r,i,j)$ as the influence of a constraint r on label j for variable i. Its formal definition requires some previous steps:

DEF: Constraint. A constraint r consists of a compatibility value C_r and its associated set of pairs variable–label. The compatibility values can be restricted to a certain interval (e.g. $[0,1]$, $[-1,1]$, $[0,+\infty]$ ...), or not restricted at all.

A constraint expresses a how compatible is a given combination of variable labels. It can be written as follows:

$$C_r \quad [(v_{i_1}, t_{j_1}^{i_1}), \ldots, (v_{i_{n_r}}, t_{j_{n_r}}^{i_{n_r}})]$$

where $1 \leq i_1, \ldots, i_{n_r} \leq N$ and
$1 \leq j_k \leq m_{i_k}$ for $k = 1 \ldots n_r$

where n_r is the constraint *degree*, that is, the number of pairs variable–label it involves, and $(v_{i_1}, t_{j_1}^{i_1}), \ldots, (v_{i_{n_r}}, t_{j_{n_r}}^{i_{n_r}})$ are the pairs involved in the constraint.

For simplicity we will note label j for variable i as t_j instead of t_j^i, since the variable i which the label is applied to is already present in the pair. The previous constraint will then be expressed as:

$$C_r \quad [(v_{i_1}, t_{j_1}), \ldots, (v_{i_{n_r}}, t_{j_{n_r}})]$$

DEF: Context weight. Obviously, the influence of a constraint on a given variable label is zero if the constraint does not include the pair variable–label. (i.e. that constraint is not applied). Then, constraints that have an influence on a given pair (v_i, t_j) are only those that include that pair, i.e., those of the form:

$$C_r \quad [(v_{i_1}, t_{j_1}), \ldots, (v_i, t_j), \ldots, (v_{i_{n_r}}, t_{j_{n_r}})]$$

We define the *context weight* for a constraint and a pair variable–label $W(r,i,j)$ as the product of the current weights for the labels appearing in the constraint except (v_i, t_j), or, if preferred, as though the weight for that label was 1.

The *context weight* states *how applicable* the constraint is given the current context of (v_i, t_j). The *constraint compatibility value* C_r states *how compatible* the pair is with the context.

Being $p_q^s(n)$ the weight assigned to label t_q for variable v_s at time n, the context weight is:

$$W(r,i,j) = p_{j_1}^{i_1}(n) \times \ldots \times p_{j_{n_r}}^{i_{n_r}}(n)$$

where $p_j^i(n)$ is not included in the product.

DEF: Constraint Influence. Once we have defined the constraint compatibility values and the context weight, we can define the influence of a constraint on the pair (v_i, t_j) as:

$$Inf(r, i, j) = C_r \times W(r, i, j)$$

DEF: Support. Once we have computed the influence for each constraint on the given label of a variable, we can compute the total support received by that label combining the influences of all constraints.

Several support functions are used in the literature, depending on the problem addressed, to define the support S_{ij} received by label j of variable i. Different support functions correspond to different ways of combining constraint influences. See [KF86] for further details on different possible support functions.

In our case, we will be using the following formula, which computes the support for a label adding the influences obtained from each constraints. Depending on the nature of the compatibility values, support values may be *negative* indicating *incompatibility*.

$$S_{ij} = \sum_r Inf(r, i, j) \tag{1}$$

2.1.2 Updating Function

The algorithm also needs to compute which is the new weight for a variable label, and this computation must be done in such a way that it can be proven to meet a certain convergence criterion, at least under appropriate conditions[1] [ZKH78, ZLM81, HZ83].

This is called the *updating function* and it is used to compute and normalize the new weights for each possible label.

Several formulas have been proposed [RHZ76], and some of them have been proven to be approximations of a gradient step algorithm.

The updating formulas must increase the weight associated with labels with a higher support, and decrease those of labels with lower support. This is achieved by multiplying the current weight of a label by a factor depending on the support received by that label. Normalization is performed in order that the weights for all the labels of a variable add up to one.

For the application describedin this paper, we will be using the following updating function, which increases the weight for a label when S_{ij} is positive and decreases it when S_{ij} is negative. Values for S_{ij} must be in $[-1, 1]$.

$$p_j^i(n+1) = \frac{p_j^i(n) \times (1 + S_{ij})}{\sum_{k=1}^{m_i} p_k^i(n) \times (1 + S_{ik})} \tag{2}$$

Since the support values S_{ij} are computed using the constraint compatibility values C_r, which may be unbounded, they do not necessarily belong to the intervals

[1]Convergence has been proven under certain conditions, but in a complex application such as POS tagging we will find cases where it is not necessarily achieved.

required by any of the above updating functions. Even in the case that the C_r were bounded, if the support computation used is additive (such as 1), the final support result can not be guaranteed to be in the required interval. Thus, it will be necessary to normalize the final support value for each label, in order to fit in the appropriate interval.

See [KI85, Tor89] for clear expositions of what is relaxation labelling and what kinds of relaxation can we get by combining different support and updating functions.

2.1.3 Convergence and Stopping Criteria

Relaxation labelling is an iterative algorithm which has been proven to converge under certain conditions [ZKH78, ZLM81, HZ83]. These conditions often require simple models –e.g. consisting only on binary constraints which must be symmetric– which are not likely to hold in complex applications such as those of NLP.

In addition, relaxation algorithms are often stopped before convergence, since they either produce better results at early iterations [RLS81, Llo83] or it is not necessary to wait until convergence to know what the result will be [ZLM81]. Different stopping criteria can be found in the literature, although most of them have a strong *ad-hoc* flavour [ER78, Pel79]. [Har83] presents a conditional probability interpretation of relaxation labelling which enables a theoretically grounded stopping criterion, unfortunately, it is only applicable in specific cases (binary constraints only, with bounded weight sum for all constraints affecting the same variable).

2.2 Application to taxonomy matching

As described in previous sections, the problem we are dealing with is to match two (possibly non–fully–connected) taxonomies. That is:

- We have a Spanish taxonomy (which, in our case, was automatically extracted from a monolingual dictionary [RRA98]). It is not fully connected –it can be seen as a set of smaller taxonomies–, and it is a a *word* taxonomy –there are no conceptual classifications such as *concept*, *sense* or *synset*.
- We have a conceptual taxonomy (e.g. WordNet), in which the nodes represent ideas such as *concepts*, *senses* or *synsets*.
- We want to relate both taxonomies in order to have a semantic assignation to each *word*–node of the Spanish taxonomy.

We will try to achieve our goal by using the relaxation labelling algorithm to assing to each Spanish word a node in the conceptual taxonomy (WordNet in our case).

The modelling of the problem is the following:

- Each word in the Spanish taxonomy is a variable for the relaxation problem.
- The possible values of that variable, are all the WN synsets which contain a word that is a possible translation of the Spanish word. Thus, we will need a bilingual dictionary to know all the possible translations for a given word.

- The relaxation algorithm will select one label for each variable, that is, one synset for each Spanish word. In that way, we will have the conceptual assignation for our Spanish taxonomy.
- The algorithm will need constraints stating when a synset is or is not a suitable assignment for a word. These constraints will rely on the taxonomy structure (e.g., a synset S_1 may be selected for a word W_1 if that word has a hyperonym W_2 in the Spanish taxonomy which may be (or is) assigned to a synset S_2 that is in turn hypernonym of S_1). More detail on the used constraints is given in section 3.

3 The Constraints

Constraint are used in relaxation labeling to increase or decrease the weight for a variable label. In our application, each word in the Spanish taxonomy is a variable, and each of its possible connections to a WordNet sysnet is a label. In this way, constraints increase the weights for the connections between a node in the Spanish taxonomy and a WordNet synset. Increasing the weight for a connection implies decreasing the weights for all the other possible connections for the same node.

To increase the weight for a connection between a Spanish taxonomy node (N_s) and a WordNet synset (N_e), we look for already connected nodes that have the same relationships in both taxonomies. For instance, that N_s has an hyponim H_s in the Spanish taxonomy and N_e has an hyponim H_e in WordNet, such that H_s is connected to H_e.

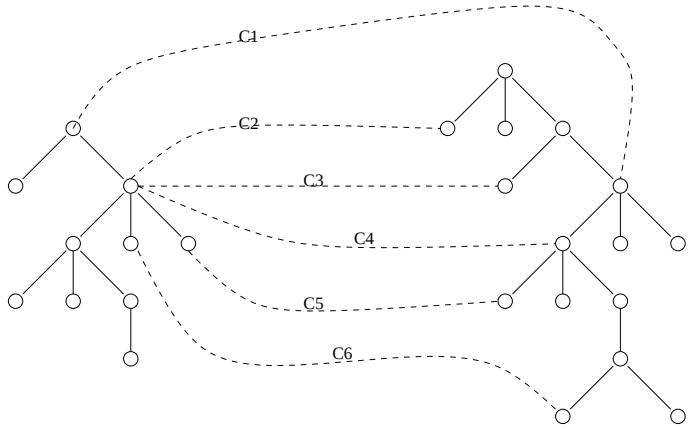

Figure 1: Example of connections between taxonomies.

Although there is a wide range of relationships between WordNet nodes which can be used to bulid constraints, we have focused on the hyper/hyponym relationships. That is, we increase the weight for a connection when the involved nodes have hyperonyms (hyponyms) also connected. We consider hyper/hyponim 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.

We distinguish different kinds of constraints, depending on whether we consider hyponyms, hyperonyms or both, on whether we consider those relationships direct or indirect, and on in which of both taxonomies we do so.

Each of the constraints described in the next sections can be used alone or combined with others.

To illustrate and describe each kind of constraints, we will use as an example the following simple taxonomy:

```
animal
    ave
        faisan
        rapaz
```

All the possible connections to WordNet for each word in the sample taxonomy are shown below. For each connection, the synset number, the WN semantic file and some synset words are shown.

```
animal ⟹(00008030 Tops <animal,animate_being,...>)
       ⟹(05957021 person <beast,brute,...>)
       ⟹(06061413 person <dunce,blockhead,...>)
       ave ⟹(00884285 animal <bird>)
           ⟹(01146542 animal <fowl,poultry,...>)
           ⟹(03073246 artifact <bird,shuttle,...>)
           ⟹(04891638 food <fowl,poultry,...>)
           ⟹(06035118 person <dame,doll,...>)
           faisan ⟹(01158294 animal <pheasant>)
                  ⟹(04893480 food <pheasant>)
           rapaz ⟹(00980561 animal <bird_of_prey,raptor,...>)
                 ⟹(05971784 person <cub,lad,...>)
                 ⟹(05992182 person <chap,fellow,...>)
                 ⟹(06110874 person <lass,young_girl,...>)
```

Connections were obtained from a bilingual dictionary and thus may contain noise, since we are connecting Spanish words to English word senses. See section 4 for details.

In the following sections, all used constraints will be described. Constraint are named 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 Spanish taxonomy: only for *immediate* nodes (I) or for *any* (A) ancestor/descendant. The second character (Y) codes the same information for the WordNet side. The third character indicates whether the constraints requires the existence of a connected hypernonym (E), hyponym (O), or both (B).

3.1 IIE constraint

The simplest constraint is to check whether the connected nodes have respective direct hyperonyms also connected. IIE stands for immediate spanish (I), immediate WordNet (I) hyperonym (E).

This constraint will increase the weights for those connections in which the immediate hyperonym of the Spanish word is connected with the immediate hyperonym of the WordNet synset. It can be graphically represented as shown in figure 2.

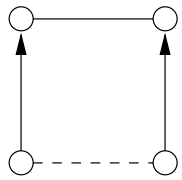

Figure 2: IIE constraint.

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

The following are the results obtained by the relaxation labeling process using IIE constraint when applied to the sample taxonomy presented above. The right hand side figures indicate the weight assigned by the relaxation algorithm to each connection.

```
animal ⟹(00008030 Tops <animal,animate_being,...>) 0.368
       ⟹(05957021 person <beast,brute,...>)         0.264
       ⟹(06061413 person <dunce,blockhead,...>)     0.368
       ave   ⟹(00884285 animal <bird>)              0.2
             ⟹(01146542 animal <fowl,poultry,...>)  0.2
             ⟹(03073246 artifact <bird,shuttle,...>) 0.2
             ⟹(04891638 food <fowl,poultry,...>)    0.2
             ⟹(06035118 person <dame,doll,...>)     0.2
             faisan ⟹(01158294 animal <pheasant>) 0.000
                    ⟹(04893480 food <pheasant>)   0.999
             rapaz ⟹(00980561 animal <bird_of_prey,raptor,...>) 0.999
                   ⟹(05971784 person <cub,lad,...>)              0.000
                   ⟹(05992182 person <chap,fellow,...>)          0.000
                   ⟹(06110874 person <lass,young_girl,...>)      0.000
```

The word *animal* is not affected by the constraint, since it has no hyperonyms. The final weights for its connections are the initial values (see section 4). The word *ave* is not modified either and also keeps its initial values. The cause in this case is that there is no synset with a connection with *ave* that has an immediate hyperonym in WN with a connection to *animal* (there are intermediate synsets in WordNet).

The weight selection for *faisán* is wrong because (04891638 `food` <*fowl,poultry,...*>) is the immediate hyperonym of (04893480 `food` <*pheasant*>), while there are two synsets between (01158294 `animal` <*pheasant*>) and its ancestor (00884285 `animal` <*bird*>).

The selection for the word *rapaz* is the right one, since the IIE constraint detects that the immediate hyperonym of (00980561 `animal` <*bird-of-prey,raptor,...*>) is the synset (00884285 `animal` <*bird*>), which is connected to *ave*.

3.2 IIO constraint

This constraint increases the weight for that connections in which an immediate hyponym of the Spanish word is connected to an immediate hyponym of the WN synset. It is represented grafically in figure 3.

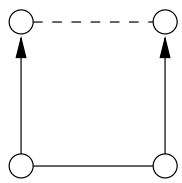

Figure 3: IIO constraint.

The results obtained using the IIO constraint are the following:

```
animal ⟹(00008030 Tops <animal,animate_being,...>) 0.368
       ⟹(05957021 person <beast,brute,...>)         0.264
       ⟹(06061413 person <dunce,blockhead,...>)     0.368
       ave   ⟹(00884285 animal <bird>)              0.004
             ⟹(01146542 animal <fowl,poultry,...>)  0.000
             ⟹(03073246 artifact <bird,shuttle,...>) 0.000
             ⟹(04891638 food <fowl,poultry,...>)    0.996
             ⟹(06035118 person <dame,doll,...>)     0.000
             faisan ⟹(01158294 animal <pheasant>) 0.5
                    ⟹(04893480 food <pheasant>)   0.5
             rapaz ⟹(00980561 animal <bird_of_prey,raptor,...>) 0.291
                   ⟹(05971784 person <cub,lad,...>)              0.209
                   ⟹(05992182 person <chap,fellow,...>)          0.209
                   ⟹(06110874 person <lass,young_girl,...>)      0.291
```

In this case, no modification is performed for the word *animal* because there are no connections between immediate hyponyms. The weight selection for *ave* is wrong because the connection of the hyponym (04893480 `food` <*pheasant*>) is stronger (0.5 *vs.* 0.291) than those caused by the hyponym (00980561 `animal` <*bird-of-prey,raptor,...*>).

The weight assignments for the words *faisán* and *rapaz* are not modified by this constraint since they do not have any hyponym.

3.3 Combining IIE and IIO

If we use both constraints at the same time, weights will be modified for words matching any of the constraints. That is, we are additively combining both constraints. In the case where both of them apply, their effects will be added. If they have opposite effects, they will cancel each other.

The results for this combination on the sample taxonomy are:

```
animal ⟹(00008030 Tops <animal,animate_being,...>) 0.368
       ⟹(05957021 person <beast,brute,...>)         0.264
       ⟹(06061413 person <dunce,blockhead,...>)     0.368
       ave   ⟹(00884285 animal <bird>)              0.336
             ⟹(01146542 animal <fowl,poultry,...>)  0.000
             ⟹(03073246 artifact <bird,shuttle,...>) 0.000
             ⟹(04891638 food <fowl,poultry,...>)    0.664
             ⟹(06035118 person <dame,doll,...>)     0.000
             faisan ⟹(01158294 animal <pheasant>) 1.49e-08
                    ⟹(04893480 food <pheasant>)   1
             rapaz ⟹(00980561 animal <bird_of_prey,raptor,...>) 1
                   ⟹(05971784 person <cub,lad,...>)          1.07e-08
                   ⟹(05992182 person <chap,fellow,...>)      1.07e-08
                   ⟹(06110874 person <lass,young_girl,...>)  1.49e-08
```

We can observe here how the word *ave* does not get a clear assignation to a unique synset. This is caused by its two hyponyms (one `food`, one `animal`) which provide contradictory information to the node.

3.4 IIB constraint

This constraint increases the weight for the connections in which the immediate hyperonym of the Spanish word is connected to the immeditate hyperonym of the WN synset and an immediate hyponym of the Spanish word is connected to an immediate hyponym of the WN synset. It is represented grafically in figure 4.

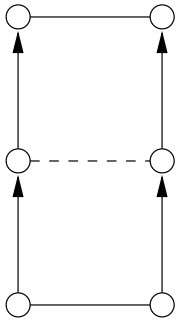

Figure 4: IIB constraint.

Note that this constraint is different than the combination of IIE and IIO. In this case we are combining them in a multiplicative fashion. That is, both of them must be satisfied to be applied.

In the sample taxonomy, this constraint would not be applied, since there is no node that has at the same time a hyperonym and a hyponym connected to WN synsets with the same hierarchy pattern.

Thus, to illustrate how this constraint works, we will assume that the Spanish taxonomy has a node *vertebrado* between *animal* and *ave*. The results for this case are the following:

animal $\Longrightarrow$(00008030 `Tops` <*animal,animate_being,...*>) 0.368
$\Longrightarrow$(05957021 `person` <*beast,brute,...*>) 0.264
$\Longrightarrow$(06061413 `person` <*dunce,blockhead,...*>) 0.368
vertebrado $\Longrightarrow$(00854210 `animal` <*vertebrate*>) 1.0
ave $\Longrightarrow$(00884285 `animal` <*bird*>) 0.999
$\Longrightarrow$(01146542 `animal` <*fowl,poultry,...*>) 0.000
$\Longrightarrow$(03073246 `artifact` <*bird,shuttle,...*>) 0.000
$\Longrightarrow$(04891638 `food` <*fowl,poultry,...*>) 0.000
$\Longrightarrow$(06035118 `person` <*dame,doll,...*>) 0.000
faisan $\Longrightarrow$(01158294 `animal` <*pheasant*>) 0.5
$\Longrightarrow$(04893480 `food` <*pheasant*>) 0.5
rapaz $\Longrightarrow$(00980561 `animal` <*bird_of_prey,raptor,...*>) 0.291
$\Longrightarrow$(05971784 `person` <*cub,lad,...*>) 0.209
$\Longrightarrow$(05992182 `person` <*chap,fellow,...*>) 0.209
$\Longrightarrow$(06110874 `person` <*lass,young_girl,...*>) 0.291

In this case, the connection selected for *ave* is the right one. It was the only connection that satisfied the IIB constraint.

3.5 AIE constraint

This constraint increases the weight for the connections in which an ancestor of the Spanish word is connected to the immeditate hyperonym of the WN synset. Its graphical representation is shown in figure 5.

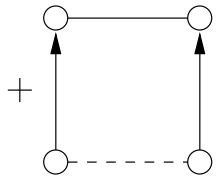

Figure 5: AIE constraint.

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

To illustrate this case, we will assume that there are two more nodes in the Spanish taxonomy and that they do not have any connection to WN (i.e. they didn't appear in the bilingual dictionary used to find the connections). These nodes will be *ovíparo*, between *vertebrado* and *ave*, and *ave_mediana*, between *ave* and *faisán*–rapaz.

In this way we will check how the constraint behaves when the nodes *faisán* and rapaz, that are no immediate hyponyms of *ave*, and *vertebrado* is not its immediate hyperonym.

The obtained results are:

```
animal ⟹(00008030 Tops <animal,animate_being,...>) 0.368
       ⟹(05957021 person <beast,brute,...>)         0.264
       ⟹(06061413 person <dunce,blockhead,...>)     0.368
       vertebrado ⟹(00854210 animal <vertebrate>) 1.0
           oviparo ⟹
               ave ⟹(00884285 animal <bird>)                0.999
                   ⟹(01146542 animal <fowl,poultry,...>)    0.000
                   ⟹(03073246 artifact <bird,shuttle,...>) 0.000
                   ⟹(04891638 food <fowl,poultry,...>)      0.000
                   ⟹(06035118 person <dame,doll,...>)       0.000
                    ave_mediana ⟹
                        faisan ⟹(01158294 animal <pheasant>) 0.000
                               ⟹(04893480 food <pheasant>)    0.999
                        rapaz ⟹(00980561 animal <bird_of_prey,raptor,...>) 0.999
                              ⟹(05971784 person <cub,lad,...>)              0.000
                              ⟹(05992182 person <chap,fellow,...>)          0.000
                              ⟹(06110874 person <lass,young_girl,...>)      0.000
```

This constraint increased the weights for the connections for words *ave*, *faisán* and *rapaz* in the same way than constraint IIE, presented in section 3.1. The difference is that IIE would not have been applied here, since the hyperonymy relationships were not immediate.

3.6 AIO constraint

This constraint increases the weight for the connections in which a descendant of the Spanish word is connected to an immediate hyponym of the WN synset. Its graphical representation is shown in figure 6.

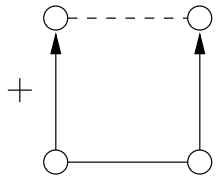

Figure 6: AIO constraint.

With the same example than in the previous case, we obtain results:

animal $\Longrightarrow$(00008030 `Tops` <*animal,animate_being,...*>) 0.368
$\Longrightarrow$(05957021 `person` <*beast,brute,...*>) 0.264
$\Longrightarrow$(06061413 `person` <*dunce,blockhead,...*>) 0.368
vertebrado $\Longrightarrow$(00854210 `animal` <*vertebrate*>) 1.0
oviparo $\Longrightarrow$
ave $\Longrightarrow$(00884285 `animal` <*bird*>) 0.003
$\Longrightarrow$(01146542 `animal` <*fowl,poultry,...*>) 0.000
$\Longrightarrow$(03073246 `artifact` <*bird,shuttle,...*>) 0.000
$\Longrightarrow$(04891638 `food` <*fowl,poultry,...*>) 0.996
$\Longrightarrow$(06035118 `person` <*dame,doll,...*>) 0.000
ave_mediana $\Longrightarrow$
faisan $\Longrightarrow$(01158294 `animal` <*pheasant*>) 0.5
$\Longrightarrow$(04893480 `food` <*pheasant*>) 0.5
rapaz $\Longrightarrow$(00980561 `animal` <*bird_of_prey,raptor,...*>) 0.291
$\Longrightarrow$(05971784 `person` <*cub,lad,...*>) 0.209
$\Longrightarrow$(05992182 `person` <*chap,fellow,...*>) 0.209
$\Longrightarrow$(06110874 `person` <*lass,young_girl,...*>) 0.291

The behaviour here is the same that we would obtain with constraint IIO if there were no intermediate nodes.

3.7 AIB constraint

This constraint increases the weight for the connections in which an ancestor of the Spanish word is connected to the immediate hyperonym of the WN synset and a descendant of the Spanish word is connected to an immediate hyponym of the WN synset. Its graphical representation is shown in figure 7.

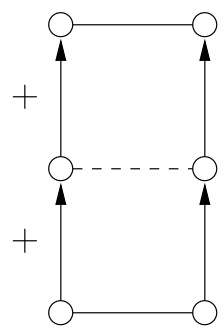

Figure 7: AIB constraint.

The results for the example in this case are:

animal $\Longrightarrow$(00008030 `Tops` <*animal,animate_being,...*>) 0.368
$\Longrightarrow$(05957021 `person` <*beast,brute,...*>) 0.264
$\Longrightarrow$(06061413 `person` <*dunce,blockhead,...*>) 0.368
vertebrado $\Longrightarrow$(00854210 `animal` <*vertebrate*>) 1.0

oviparo ⟹
ave ⟹(00884285 `animal` <*bird*>) 0.999
⟹(01146542 `animal` <*fowl,poultry,...*>) 0.000
⟹(03073246 `artifact` <*bird,shuttle,...*>) 0.000
⟹(04891638 `food` <*fowl,poultry,...*>) 0.000
⟹(06035118 `person` <*dame,doll,...*>) 0.000
ave_mediana ⟹
faisan ⟹(01158294 `animal` <*pheasant*>) 0.5
⟹(04893480 `food` <*pheasant*>) 0.5
rapaz ⟹(00980561 `animal` <*bird_of_prey,raptor,...*>) 0.291
⟹(05971784 `person` <*cub,lad,...*>) 0.209
⟹(05992182 `person` <*chap,fellow,...*>) 0.209
⟹(06110874 `person` <*lass,young_girl,...*>) 0.291

The algorithm modifies here the connections for *ave*, assigning the right synset.

3.8 Combining AIE, AIO and AIB.

If we use constraints AIE, AIO and AIB simultaneously, in additive combination, be obtain the results presented below. Note that in this case we apply either a hyperonym 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 hyperonym and hyponym will have their weights doubly increased.

animal ⟹(00008030 `Tops` <*animal,animate_being,...*>) 0.368
⟹(05957021 `person` <*beast,brute,...*>) 0.264
⟹(06061413 `person` <*dunce,blockhead,...*>) 0.368
vertebrado ⟹(00854210 `animal` <*vertebrate*>) 1.0
oviparo ⟹
ave ⟹(00884285 `animal` <*bird*>) 0.998
⟹(01146542 `animal` <*fowl,poultry,...*>) 0.000
⟹(03073246 `artifact` <*bird,shuttle,...*>) 0.000
⟹(04891638 `food` <*fowl,poultry,...*>) 0.001
⟹(06035118 `person` <*dame,doll,...*>) 0.000
ave_mediana ⟹
faisan ⟹(01158294 `animal` <*pheasant*>) 0.000
⟹(04893480 `food` <*pheasant*>) 1.0
rapaz ⟹(00980561 `animal` <*bird_of_prey,raptor,...*>) 1.0
⟹(05971784 `person` <*cub,lad,...*>) 0.000
⟹(05992182 `person` <*chap,fellow,...*>) 0.000
⟹(06110874 `person` <*lass,young_girl,...*>) 0.000

All connections suffer weight modification except *animal*. Again, *faisán* gets the wrong connection assignment.

3.9 IAE constraint

This constraint increases the weight for the connections in which the immediate hyperonym of the Spanish word is connected to an ancestor of the WN synset. Its graphical representation is shown in figure 8.

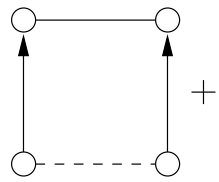

Figure 8: IAE constraint.

The results obtained with this constraint are the following:

animal $\Longrightarrow$(00008030 `Tops` <*animal,animate_being,...*>) 0.368
$\Longrightarrow$(05957021 `person` <*beast,brute,...*>) 0.264
$\Longrightarrow$(06061413 `person` <*dunce,blockhead,...*>) 0.368
ave $\Longrightarrow$(00884285 `animal` <*bird*>) 0.98
$\Longrightarrow$(01146542 `animal` <*fowl,poultry,...*>) 0.02
$\Longrightarrow$(03073246 `artifact` <*bird,shuttle,...*>) 0.000
$\Longrightarrow$(04891638 `food` <*fowl,poultry,...*>) 0.000
$\Longrightarrow$(06035118 `person` <*dame,doll,...*>) 0.000
faisan $\Longrightarrow$(01158294 `animal` <*pheasant*>) 1.0
$\Longrightarrow$(04893480 `food` <*pheasant*>) 0.000
rapaz $\Longrightarrow$(00980561 `animal` <*bird_of_prey,raptor,...*>) 1.0
$\Longrightarrow$(05971784 `person` <*cub,lad,...*>) 0.000
$\Longrightarrow$(05992182 `person` <*chap,fellow,...*>) 0.000
$\Longrightarrow$(06110874 `person` <*lass,young_girl,...*>) 0.000

In this case the weights for connections for words *faisán* and rapaz yield the correct assignment, since the constraint has been applied to the word *ave* and its hypernonym *animal*, even when there are two intermediate nodes between (00008030 `Tops` <*animal,animate_being,...*>) and (00884285 `animal` <*bird*>) in WordNet.

3.10 IAO constraint

This constraint increases the weight for the connections in which an immediate hyponym of the Spanish word is connected to a descendant of the WN synset. Its graphical representation is shown in figure 9.

animal $\Longrightarrow$(00008030 `Tops` <*animal,animate_being,...*>) 1.0
$\Longrightarrow$(05957021 `person` <*beast,brute,...*>) 0.000
$\Longrightarrow$(06061413 `person` <*dunce,blockhead,...*>) 0.000
ave $\Longrightarrow$(00884285 `animal` <*bird*>) 0.995
$\Longrightarrow$(01146542 `animal` <*fowl,poultry,...*>) 0.000

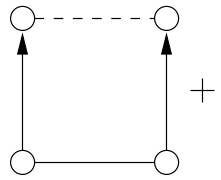

Figure 9: IAO constraint.

$\Longrightarrow$(03073246 `artifact` <*bird,shuttle,...*>) 0.000
$\Longrightarrow$(04891638 `food` <*fowl,poultry,...*>) 0.005
$\Longrightarrow$(06035118 `person` <*dame,doll,...*>) 0.000
faisan $\Longrightarrow$(01158294 `animal` <*pheasant*>) 0.5
$\Longrightarrow$(04893480 `food` <*pheasant*>) 0.5
rapaz $\Longrightarrow$(00980561 `animal` <*bird_of_prey,raptor,...*>) 0.291
$\Longrightarrow$(05971784 `person` <*cub,lad,...*>) 0.209
$\Longrightarrow$(05992182 `person` <*chap,fellow,...*>) 0.209
$\Longrightarrow$(06110874 `person` <*lass,young_girl,...*>) 0.291

In this case the constraint is also applied to *animal*, though there are two nodes in WN between (00008030 `Tops` <*animal,animate_being,...*>) and (00884285 `animal` <*bird*>). The constraint recurses downwards WordNet searching for connected descendants.

The weight assignment for *animal* and *ave* select the right connection. In the case of *ave*, despite the word *faisán* has a nearer connected descendant for `food` than for `animal`, the `animal` sense is chosen because of the contribution of *rapaz*.

3.11 IAB constraint

This constraint increases the weight for the connections in which the immediate hyperonym of the Spanish word is connected to an ancestor of the WN synset and an immediate hyponym of the Spanish word is connected to a descendant of the WN synset. Its graphical representation is presented in figure 10.

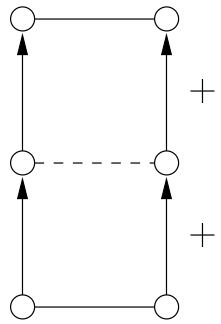

Figure 10: IAB constraint.

The obtained results are:

animal $\Longrightarrow$(00008030 `Tops` <*animal,animate_being,...*>) 0.368
$\Longrightarrow$(05957021 `person` <*beast,brute,...*>) 0.264
$\Longrightarrow$(06061413 `person` <*dunce,blockhead,...*>) 0.368
ave $\Longrightarrow$(00884285 `animal` <*bird*>) 0.999
$\Longrightarrow$(01146542 `animal` <*fowl,poultry,...*>) 0.000
$\Longrightarrow$(03073246 `artifact` <*bird,shuttle,...*>) 0.000
$\Longrightarrow$(04891638 `food` <*fowl,poultry,...*>) 0.000
$\Longrightarrow$(06035118 `person` <*dame,doll,...*>) 0.000
faisan $\Longrightarrow$(01158294 `animal` <*pheasant*>) 0.5
$\Longrightarrow$(04893480 `food` <*pheasant*>) 0.5
rapaz $\Longrightarrow$(00980561 `animal` <*bird_of_prey,raptor,...*>) 0.291
$\Longrightarrow$(05971784 `person` <*cub,lad,...*>) 0.209
$\Longrightarrow$(05992182 `person` <*chap,fellow,...*>) 0.209
$\Longrightarrow$(06110874 `person` <*lass,young_girl,...*>) 0.291

The constraint produces a right weight assignment for ave, combining the constraints for *animal–faisán* (`animal`), and for *animal–rapaz* (`animal`)

3.12 Combining IAE, IAO and IAB

Using simultaneously constraints IAE, IAO and IAB we obtain the results presented below. As in section 3.8, the combination produces a stronger evidence when both connected hyperonym and hyponym are found.

animal $\Longrightarrow$(00008030 `Tops` <*animal,animate_being,...*>) 1.0
$\Longrightarrow$(05957021 `person` <*beast,brute,...*>) 0.000
$\Longrightarrow$(06061413 `person` <*dunce,blockhead,...*>) 0.000
ave $\Longrightarrow$(00884285 `animal` <*bird*>) 0.999
$\Longrightarrow$(01146542 `animal` <*fowl,poultry,...*>) 0.000
$\Longrightarrow$(03073246 `artifact` <*bird,shuttle,...*>) 0.000
$\Longrightarrow$(04891638 `food` <*fowl,poultry,...*>) 0.000
$\Longrightarrow$(06035118 `person` <*dame,doll,...*>) 0.000
faisan $\Longrightarrow$(01158294 `animal` <*pheasant*>) 0.999
$\Longrightarrow$(04893480 `food` <*pheasant*>) 0.000
rapaz $\Longrightarrow$(00980561 `animal` <*bird_of_prey,raptor,...*>) 1.0
$\Longrightarrow$(05971784 `person` <*cub,lad,...*>) 0.000
$\Longrightarrow$(05992182 `person` <*chap,fellow,...*>) 0.000
$\Longrightarrow$(06110874 `person` <*lass,young_girl,...*>) 0.000

In this example, all the selected connections are the right ones.

3.13 AAE constraint

This constraint increases the weight for the connections in which an ancestor of the Spanish word is connected to an ancestor of the WN synset. Its graphical representation is presented in figure 11.

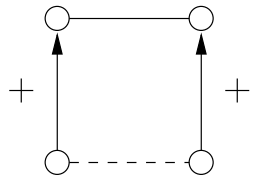

Figure 11: AAE constraint.

The constraint affects all nodes that have a hyponym. The results obtained applying this constraint are the following:

animal $\Longrightarrow$(00008030 `Tops` <*animal,animate_being,...*>) 0.368
$\Longrightarrow$(05957021 `person` <*beast,brute,...*>) 0.264
$\Longrightarrow$(06061413 `person` <*dunce,blockhead,...*>) 0.368
vertebrado $\Longrightarrow$(00854210 `animal` <*vertebrate*>) 1.0
oviparo $\Longrightarrow$
ave $\Longrightarrow$(00884285 `animal` <*bird*>) 0.98
$\Longrightarrow$(01146542 `animal` <*fowl,poultry,...*>) 0.02
$\Longrightarrow$(03073246 `artifact` <*bird,shuttle,...*>) 0.000
$\Longrightarrow$(04891638 `food` <*fowl,poultry,...*>) 0.000
$\Longrightarrow$(06035118 `person` <*dame,doll,...*>) 0.000
ave_mediana $\Longrightarrow$
faisan $\Longrightarrow$(01158294 `animal` <*pheasant*>) 1.0
$\Longrightarrow$(04893480 `food` <*pheasant*>) 0.000
rapaz $\Longrightarrow$(00980561 `animal` <*bird_of_prey,raptor,...*>) 1.0
$\Longrightarrow$(05971784 `person` <*cub,lad,...*>) 0.000
$\Longrightarrow$(05992182 `person` <*chap,fellow,...*>) 0.000
$\Longrightarrow$(06110874 `person` <*lass,young_girl,...*>) 0.000

3.14 AAO constraint

This constraint increases the weight for the connections in which a descendant of the Spanish word is connected to a descendant of the WN synset. Its graphical representation is presented in figure 12.

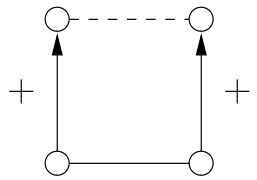

Figure 12: AAO constraint.

The constraint affects all nodes that have a hyponym. The results produced by the application of this constraint are:

animal $\Longrightarrow$(00008030 `Tops` <*animal,animate_being,...*>) 1.0
$\Longrightarrow$(05957021 `person` <*beast,brute,...*>) 0.000
$\Longrightarrow$(06061413 `person` <*dunce,blockhead,...*>) 0.000
vertebrado $\Longrightarrow$(00854210 `animal` <*vertebrate*>) 1.0
oviparo $\Longrightarrow$
ave $\Longrightarrow$(00884285 `animal` <*bird*>) 0.995
$\Longrightarrow$(01146542 `animal` <*fowl,poultry,...*>) 0.000
$\Longrightarrow$(03073246 `artifact` <*bird,shuttle,...*>) 0.000
$\Longrightarrow$(04891638 `food` <*fowl,poultry,...*>) 0.005
$\Longrightarrow$(06035118 `person` <*dame,doll,...*>) 0.000
ave_mediana $\Longrightarrow$
faisan $\Longrightarrow$(01158294 `animal` <*pheasant*>) 0.5
$\Longrightarrow$(04893480 `food` <*pheasant*>) 0.5
rapaz $\Longrightarrow$(00980561 `animal` <*bird_of_prey,raptor,...*>) 0.291
$\Longrightarrow$(05971784 `person` <*cub,lad,...*>) 0.209
$\Longrightarrow$(05992182 `person` <*chap,fellow,...*>) 0.209
$\Longrightarrow$(06110874 `person` <*lass,young_girl,...*>) 0.291

3.15 AAB constraint

This constraint increases the weight for the connections in which an ancestor of the Spanish word is connected to an ancestor of the WN synset and a descendant of the Spanish word is connected to a descendant of the WN synset. Its graphical representation is presented in figure 13.

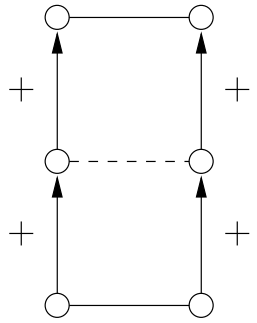

Figure 13: AAB constraint.

The obtained results are:

animal $\Longrightarrow$(00008030 `Tops` <*animal,animate_being,...*>) 0.368
$\Longrightarrow$(05957021 `person` <*beast,brute,...*>) 0.264
$\Longrightarrow$(06061413 `person` <*dunce,blockhead,...*>) 0.368
vertebrado $\Longrightarrow$(00854210 `animal` <*vertebrate*>) 1.0
oviparo $\Longrightarrow$
ave $\Longrightarrow$(00884285 `animal` <*bird*>) 0.999
$\Longrightarrow$(01146542 `animal` <*fowl,poultry,...*>) 0.000

$\Longrightarrow$(03073246 `artifact` <*bird,shuttle,...*>) 0.000
$\Longrightarrow$(04891638 `food` <*fowl,poultry,...*>) 0.000
$\Longrightarrow$(06035118 `person` <*dame,doll,...*>) 0.000
ave_mediana $\Longrightarrow$
faisan $\Longrightarrow$(01158294 `animal` <*pheasant*>) 0.5
$\Longrightarrow$(04893480 `food` <*pheasant*>) 0.5
rapaz $\Longrightarrow$(00980561 `animal` <*bird_of_prey,raptor,...*>) 0.291
$\Longrightarrow$(05971784 `person` <*cub,lad,...*>) 0.209
$\Longrightarrow$(05992182 `person` <*chap,fellow,...*>) 0.209
$\Longrightarrow$(06110874 `person` <*lass,young_girl,...*>) 0.291

The only affected node is *ave*, since it is the only node with connected ancestor and descendant.

3.16 Combining AAE, AAO and AAB

If we apply, as in previous cases, the combination of constraint AAE,AAO and AAB, we obtain:

animal $\Longrightarrow$(00008030 `Tops` <*animal,animate_being,...*>) 0.949
$\Longrightarrow$(05957021 `person` <*beast,brute,...*>) 0.021
$\Longrightarrow$(06061413 `person` <*dunce,blockhead,...*>) 0.029
vertebrado $\Longrightarrow$(00854210 `animal` <*vertebrate*>) 1.0
oviparo $\Longrightarrow$
ave $\Longrightarrow$(00884285 `animal` <*bird*>) 0.873
$\Longrightarrow$(01146542 `animal` <*fowl,poultry,...*>) 0.040
$\Longrightarrow$(03073246 `artifact` <*bird,shuttle,...*>) 0.027
$\Longrightarrow$(04891638 `food` <*fowl,poultry,...*>) 0.032
$\Longrightarrow$(06035118 `person` <*dame,doll,...*>) 0.027
ave_mediana $\Longrightarrow$
faisan $\Longrightarrow$(01158294 `animal` <*pheasant*>) 0.946
$\Longrightarrow$(04893480 `food` <*pheasant*>) 0.054
rapaz $\Longrightarrow$(00980561 `animal` <*bird_of_prey,raptor,...*>) 0.929
$\Longrightarrow$(05971784 `person` <*cub,lad,...*>) 0.020
$\Longrightarrow$(05992182 `person` <*chap,fellow,...*>) 0.020
$\Longrightarrow$(06110874 `person` <*lass,young_girl,...*>) 0.029

The process afects now all nodes, since any node has either a connected ascendant or a connected descendant.

3.17 Relationships between constraints

Some of the presented constraints are particular cases of others. For instance, constraint AIE checks whether any ascendant in the Spanish taxonomy is connected to the immediate hyperonym in WordNet. The search is performed upwards, and stopped when the first match is found. The first matching ascendant could be the immediate

hyperonym in the Spanish taxonomy, which is what IIE constraint checks. Thus, IIE is a particular case of AIE.

Analogous reasonings yield that IIE is also a particular case of IAE. Both AIE and IAE are particular cases of AAE. Thus, we have the constraint relationships presented in figure 14.

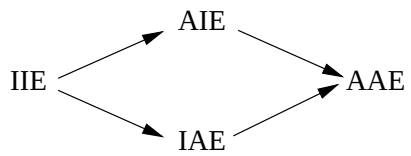

Figure 14: Particular–general relationships among hyperonym constraints.

For the constraints affecting hyponyms, we obtain the relationships in figure 15. For those affecting both hyperonyms and hyponyms we obtain the relationships in figure 16

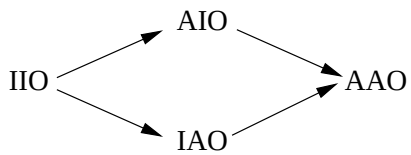

Figure 15: Particular–general relationships among hyponym constraints.

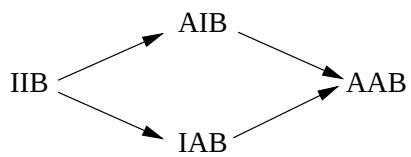

Figure 16: Particular–general relationships among hyperonym/hyponym constraints.

3.18 Grouping Constraints

There are constraints that work better combined with others. For instance, the constraint IIO is only applied to nodes that have connected hyponyms. Then, leaf nodes –which have no hyponym at all– will never be affected by that constraint. Simmetrically, constraint IIE only affects nodes which have hyperonyms, and thus will never modify top nodes in the taxonomy. Combining them both, every node in the taxonomy can be affected by at least one of the constraints. We can reinforce the effect on nodes having both a hyperonym and a hyponym, introducing also the constraint IIB in the pack –as described in section 3.8–. We will name this pack II.

In the same way, we can group AIE, AIO and AIB in a pack named AI, IAE, IAO and IAB in IA, and finally AAE, AAO and AAB in AA.

The particular–general relationships of the individual constraints are also valid for the packs, as shown in figure 17.

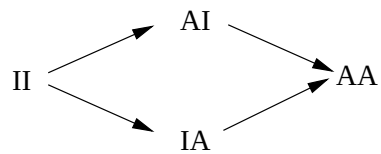

Figure 17: Particular–general relationships among constraint classes.

3.19 Weights for Constraints

When using a constraint alone, any positive value is adecuate for the compatibility value expressed by the constraint, since all applications will have the same strengh. But when we use several constraints at the same time, it becomes necessary to give each one an appropriate compatibility value, according to our needs. For instance, when we use simultaneously IIE and IIO, the later constraint increases the weight for the connection between nodes with connected hyponyms. Given that the hierarchical structure of WN is –in most cases– tree-like, we will consider the evidence supplied by a common hyponym as 1 and use this value as the compatibility value for constraint IIO. On the other hand, constraint IIE increases the weight for the connection between nodes with connected hyperonyms. It seems logical that the evidence supplied by a connected hyperonym is smaller than that supplied by a connected hyponym, since all siblings have a common hyperonym without being necessarily the same node. Since WN1.5 has 4.54 hyponyms per sysnet in average, we can assume that the evidence supplied by a connected hyperonym is $\frac{1}{4.52} \approx 0.22$, and use this value as the compatibility value for IIE.

When we use simultaneously IIE, IIO and IIB (pack II), we can use the same compatibility values for IIE (0.22) and IIO (1). We will choose the compatibility value for IIB according to the following criteria:

- The support for a connection provided by a constraint is the compatibility value for the constraint multiplied by the weights for the connections intervening in it (see section 2.1.1).

  Let n be the average number of candidate connections per word. Then, the average weight for a connection is $\frac{1}{n}$ and thus the average support contribution from constraint IIE is $0.22 \times \frac{1}{n}$. In the same way, the average support from IIO is $1 \times \frac{1}{n}$.

  Being x the compatibility value for constraint IIB, we have that its support should be $x \times \frac{1}{n^2}$ –since there we are combining two connections (a hyponym and a hyperonym) with a weight of $\frac{1}{n}$ each.

- It should be larger than the combination of the other two, since it is more informative to have simultaneously connected hyperonym and hyponym, than having one of each separately. Thus, the support provided by this constraint should be larger than the support provided by the other two toghether. The combined support for a disjunction is the addition, i.e. $(1 + 0.22) \times \frac{1}{n}$.

Thus, to find a value for x that satisfies these conditions, we must solve the

inequation

$$x \times \frac{1}{n^2} > 1.22 \times \frac{1}{n} \qquad \text{getting as a solution} \quad x > 1.22 \times n.$$

The number of connections n depends on the bilingual dictionary used to obtain them and on the number of Spanish words in the taxonomy. In our case, several tests produce values about $n \approx 8$. We chose $10 > 1.22 \times 8$ as a compatibility value for IIB constraint.

We can apply the same compatibility values to the other constraint packs. But in those groups there is another factor which is the recursive search of an ancestor or descendant in any of both taxonomies. The compatibility value for a constraint should depend on the distance of the ancestor/descendant to the affected connection, that is, it should be –according to the particular–general relationships discussed in section 3.17– the corresponding value when the ancestor/descendant is the immediate hyperonym/hyponym, but it should progressively decrease when the distance to the ancestor/descendant increases, since the supplied evidence is weaker. Thus, the non–immediate constraints have a compatibility value which starts being the same than their immediate partners, but decreases by a certain factor at each hierarchy level. For the same reasons than above, we choose this factor to be $0.22(\approx \frac{1}{4.54})$

4 Experiments and Results

In this section we the performed experiments and results obtained will be described. A brief description of the used resources is included to set the reader in the test environment.

4.1 Spanish Taxonomies

We tested the relaxation labeling algorithm with the described constraints on a sert of Spanish taxonomies automatically extracted from monolingual dictionaries [RAA97, RRA98]. The top nodes of the taxonomies were automatically assigned to a WordNet semantic file. We used in our experiments the taxonomies assigned to the files `noun.animal`, `noun.food`, `noun.cognition` and `noun.communication`.

We performed experiments directly on the taxonomies extracted by [RAA97], as well as on slight variations of them. Namely, we tested on the following *modified* taxonomies:

+top Add a new virtual top as an hyperonym of all the top nodes of taxonomies belonging to the same semantic file. The virtual top is connected to the top synset of the WordNet semantic file. In this way, many taxonomies for a semantic files, are converted to a single one. For instance, add a virtual top *animal* as a hyperonym of all tops of all taxonomies assigned to `noun.animal`

only-top Perform the tests only using those taxonomies which have a top node with the word corresponding to the translation of the top synset for the semantic file. For instance, use only those taxonomies assigned to `noun.animal` that have the word *animal* as a top.

no-senses The original taxonomies are build taking into account dictionary entries. Thus, the nodes are not words, but dictionary *senses*. This test consists of colapsing toghether all the sibling nodes that have the same word, regardless of the dictionary sense it came from. This is done as an attempt to minimize the noise introduced at the sense level by the taxonomy building procedure.

Table 1 shows the number of nodes in each test taxonomy.

	`noun.animal`	`noun.food`	`noun.cognition`	`noun.communication`
original	1675	746	1494	2241
+top	1676	747	1495	2242
only-top	441	21	122	10
no-senses	1600	696	1402	2063

Table 1: Number of nodes in each test taxonomy.

4.2 Bilingual dictionaries

The connections between a node in the Spanish taxonomy and WN synsets were derived from bilingual dictionaries. Each node is assigned all the synsets for all the words that are a possible translation for the Spanish word, according to the bilingual dictionary. Although the Spanish taxonomy nodes are dictionary entries, bilingual dictionaries translate words. Thus, this step introduces noise in the form of irrelevant connections, since not all translations necessarily hold for a single dictionary entry.

We used two different dictionaries in our experiements: On the one habd, the VOX Essential, with 66,347 translations. On the other hand, a bilingual dictionary obtained by integrating the VOX Essential plus other several bilingual sources available. This multisource dictionary contains 195,147 translations.

To select a WN synset for each node in the Spanish taxonomy we need to know which are its possible connections. The connections for a certain word can only be obtained if this word is present in the bilingual dictionary. Since not all words in the taxonomy appear in our bilingual dictionaries, we will have partial coverage of the taxonomy. Tables 2 and 3 show the number of nodes in each taxonomy that appear in the bilingual dictionary (and thus, that may be connected to WN). The figure is also given in percentage over the taxonomy size.

	`noun.animal`	`noun.food`	`noun.cognition`	`noun.communication`
original	444 (26%)	269 (36%)	521 (35%)	1088 (49%)
+top	443 (26%)	270 (36%)	522 (35%)	1089 (49%)
only-top	117 (28%)	12 (57%)	35 (29%)	9 (90%)
no-senses	383 (24%)	218 (31%)	440 (31%)	933 (45%)

Table 2: Number of nodes with bilingual connection in each test taxonomy, using VOX Essential.

	noun.animal	noun.food	noun.cognition	noun.communication
original	755 (45%)	414 (55%)	814 (54%)	1486 (66%)
+top	756 (45%)	415 (56%)	815 (55%)	1487 (66%)
only-top	187 (45%)	15 (71%)	50 (41%)	10 (100%)
no-senses	685 (43%)	359 (52%)	726 (52%)	1316 (64%)

Table 3: Number of nodes with bilingual connection in each test taxonomy, unsing the multisource dictionary.

Among the words that appear in the bilingual dictionary and thus have candidate connections to WN, some have only one candidate connection –i.e. are *monosemous*–. Since selecting a connection for these cases is trivial, we will focus on the *ambiguous* nodes, i.e. those that have more than one candidate connection. Tables 4 and 5 show the amount of ambiguous nodes in each test taxonomy. Percentage over the number of words with bilingual connection is also given.

	noun.animal	noun.food	noun.cognition	noun.communication
original	352 (80%)	213 (79%)	430 (83%)	944 (87%)
+top	353 (80%)	214 (79%)	431 (83%)	945 (87%)
only-top	78 (67%)	11 (92%)	25 (71%)	8 (89%)
no-senses	297 (78%)	170 (78%)	350 (80%)	800 (86%)

Table 4: Number of nodes with more than one candidate connection, using VOX Essential.

	noun.animal	noun.food	noun.cognition	noun.communication
original	578 (77%)	334 (81%)	604 (74%)	1289 (87%)
+top	579 (77%)	335 (81%)	605 (74%)	1290 (87%)
only-top	122 (65%)	15 (100%)	35 (70%)	9 (90%)
no-senses	514 (75%)	283 (79%)	520 (72%)	1130 (86%)

Table 5: Number of nodes with more than one candidate connection, using the multisource dictionay.

4.3 Results

The performed tests include applying all constraint packs (II, AI, IA and AA) to all taxonomies for the four test semantic files, conecting them to WN either via VOX Essential or via the multisource dictionary.

Coverage figures for every test can be found in appendix A. Coverage is computed as the amount of nodes for which some constraint is applied and thus their weight assignment is changed. Percentage is given over the total amount of nodes with bilingual connections.

Table 8 contains coverage figures for `noun.animal`. In the same way, table 9 shows coverage figures for the taxonomies in `noun.food` semantic file, and table 10

presents the values for the taxonomies in `noun.cognition`. Finally, table 11 shows coverage figures for the taxonomies in `noun.communication` semantic file.

In any case, presented results show that with a larger bilingual dictionary, there are more words with candidate connections and thus the coverage is larger.

4.3.1 Precision Results

More interesting is evaluating the precision of our algorithm. We hand checked the results for the case supposed to be the best, that is, using AA constraints and the multisource dictionary. Precision results can be divided in several cases:

T_{OK}, F_{OK} The Spanish taxonomy is well build and correctly assigned to the semantic file.

T_{OK}, F_{NOK} The Spanish taxonomy is well build, but wrongly assigned to the semantic file.

T_{NOK} The Spanish taxonomy is wrongly build.

In each case, the algorithm selects a connection for each node, we will count how many connections are right/wrong in the first and second cases. In the third case the taxonomy was wrongly extracted and is nonsense, so the assignations cannot be evaluated.

Note that we can distinguish right/wrong assignations in the second case because the connections are taken into account over the whole WN, not only on the semantic file being processed. So, the algorithm may end up correctly assigning the words of a hierarchy, even when it was assigned to the wrong semantic file. For instance, in the hierarchy

piel	$\Longrightarrow$(03617358 `body` <*skin, tegument* >)	0.154	
	$\Longrightarrow$(01249099 `animal` <*fur* >)	0.012	
	$\Longrightarrow$(03082323 `artifact` <*skin* >)	0.012	
	$\Longrightarrow$(04962858 `food` <*peel* >)	0.017	
	$\Longrightarrow$(08869224 `substance` <*fur, pelt* >)	0.805	
	marta $\Longrightarrow$(03465327 `attribute` <*sable,coal_back ...*>)		0.015
	$\Longrightarrow$(01759335 `animal` <*marten* >)		0.028
	$\Longrightarrow$(03016526 `artifact` <*sable* >)		0.015
	$\Longrightarrow$(08870582 `substance` <*sable*>)		0.932
	vison $\Longrightarrow$(02787720 `artifact` <*mink,mink_coat* >)		0.015
	$\Longrightarrow$(01752582 `animal` <*mink* >)		0.015
	$\Longrightarrow$(08870315 `substance` <*mink*>)		0.97

the `noun.substance` synsets for each word are selected, since there was no synset for *piel* that was ancestor of the `animal` senses of *marta* and *visón*.

In this case, the hierarchy was well build, and well solved by the algorithm. The only mistake was having assigned it to the `noun.animal` semantic file, so we will count it as a right choice of the relaxation labeling algorithm, but write it in a separate column.

Tables 6 and 7 show the precision rates for each test taxonomy. In the former, figures are given over ambiguous words (nodes with more than one candidate connection). In the later, figures are computed overall (nodes with at least one candidate

connection). Accuracy is computed to the semantic file level, i.e., if a word is assigned a synset of the right semantic file, it is computed as right, otherwise, as wrong.

	T_{OK}, F_{OK}	T_{OK}, F_{NOK}	Total T_{OK}	T_{NOK}
`animal`	279 (90%)	30 (91%)	309 (90%)	23
`food`	166 (94%)	3 (100%)	169 (94%)	2
`cognition`	198 (67%)	27 (90%)	225 (69%)	49
`communication`	533 (77%)	40 (97%)	573 (78%)	16

Table 6: Precision results over ambiguous words for the test taxonomies.

	Tax. OK & File OK	Tax. OK & File NOK	Total Tax. OK
`animal`	424 (93%)	62 (95%)	486 (93%)
`food`	166 (94%)	83 (100%)	149 (96%)
`cognition`	200 (67%)	245 (99%)	445 (82%)
`communication`	536 (77%)	234 (99%)	760 (81%)

Table 7: Precision results over all words for the test taxonomies.

5 Conclusions

We have applied the relaxation labeling algorithm to assign an appropriate WN synset to each node of an automatically extracted taxonomy. Preliminary results have been reported, and they point that this may be an accurate method to connect taxonomies, either for the same or different languages.

The experiments performed up to now seem to indicate that:

- The relaxation labeling algorithm is a good technique to link two different taxonomies. For each node with several possible connections, the candidate that best matches the surrounding structure is selected.
- There is a certain amount of noise in the different phases of the process. First, the taxonomies were automatically acquired and assigned to semantic files. Second, the bilingual dictionary translates words, not senses, which introduces irrelevant candidate connections. Using a sense–based bilingual dictionary instead of a word–based one should increase the precision of the algorithm, since now it tends to select the same connection for the same word regardless of the dictionary entry it came from.
- The size and coverage of the bilingual dictionaries used to establish the candidate connections is an important issue. A dictionary with larger coverage increases the amount of nodes with candidate connections and thus the coverage of the algorithm.

6 Further Work

Some issues to be addressed to improve the algorithm performance, and to exploit its possibilities are:

- Further test and evaluate the precision of the algorithm. In this direction we plan to perform wider hand checking of the results, as well as using the techinque to link WN1.5 with WN1.6. Since there is already a mapping between both versions, the experiment would provide an idea of the accuracy of the thecnique and of its aplicability to large taxonomies.
- 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.
- To palliate the scarce coverage of the bilingual dictionaries, candidate connections could be inferred from connections of surrounding nodes. For instance, if a node has no candidate connections, but its hyperonym does, we could consider as candidate connections for that node all the hyponyms of the synset connected to its hyperonym.
- Use the algorithm to enrich the Spanish part of EuroWordNet taxonomy. It could also be applied to include taxonomies for other languages not currently in the EWN project.

7 Acknowledgements

This research has been partially funded by the Spanish Research Department (ITEM Project TIC96-1243-C03-03), the Catalan Research Department (CREL project), and the UE Comision (EuroWordNet LE4003).

A Coverage figures

Bilingual dict.	Taxonomy	II	AI	IA	AA
VOX essential	original	67 (19%)	67 (19%)	172 (49%)	177 (50%)
	+top	74 (21%)	75 (21%)	203 (58%)	253 (72%)
	only-top	12 (15%)	12 (15%)	60 (77%)	63 (81%)
	no-senses	59 (20%)	59 (20%)	156 (53%)	161 (54%)
Multisource	original	134 (23%)	135 (23%)	357 (62%)	365 (63%)
	+top	138 (24%)	143 (25%)	375 (65%)	454 (78%)
	only-top	17 (14%)	18 (15%)	100 (82%)	106 (87%)
	no-senses	118 (23%)	119 (20%)	311 (61%)	319 (62%)

Table 8: Coverage for `noun.animal`.

Bilingual dict.	Taxonomy	II	AI	IA	AA
VOX essential	original	57 (27%)	66 (31%)	82 (38%)	92 (43%)
	+top	61 (29%)	70 (33%)	117 (55%)	156 (73%)
	only-top	4 (36%)	4 (36%)	6 (55%)	6 (55%)
	no-senses	46 (27%)	54 (32%)	70 (41%)	78 (46%)
Multisource	original	119 (36%)	130 (39%)	164 (49%)	180 (54%)
	+top	134 (40%)	158 (47%)	194 (58%)	259 (73%)
	only-top	6 (40%)	6 (40%)	12 (80%)	13 (87%)
	no-senses	102 (36%)	111 (39%)	143 (51%)	156 (55%)

Table 9: Coverage for `noun.food`.

Bilingual dict.	Taxonomy	II	AI	IA	AA
VOX essential	original	124 (29%)	131 (30%)	208 (48%)	221 (51%)
	+top	130 (30%)	145 (34%)	264 (61%)	375 (87%)
	only-top	4 (16%)	4 (16%)	18 (72%)	19 (76%)
	no-senses	109 (31%)	115 (33%)	175 (50%)	186 (53%)
Multisource	original	225 (37%)	230 (38%)	360 (60%)	373 (62%)
	+top	230 (38%)	240 (40%)	395 (65%)	509 (84%)
	only-top	7 (20%)	7 (20%)	25 (71%)	26 (74%)
	no-senses	192 (37%)	197 (38%)	306 (59%)	318 (61%)

Table 10: Coverage for `noun.cognition`.

Bilingual dict.	Taxonomy	II	AI	IA	AA
VOX essential	original	260 (28%)	273 (29%)	391 (41%)	406 (43%)
	+top	292 (31%)	337 (36%)	481 (51%)	782 (83%)
	only-top	4 (50%)	4 (50%)	8 (100%)	8 (100%)
	no-senses	230 (29%)	243 (30%)	338 (42%)	353 (44%)
Multisource	original	552 (43%)	577 (45%)	737 (57%)	760 (59%)
	+top	589 (46%)	697 (54%)	802 (62%)	1136 (88%)
	only-top	7 (78%)	7 (78%)	9 (100%)	9 (100%)
	no-senses	485 (43%)	509 (45%)	645 (57%)	668 (59%)

Table 11: Coverage for `noun.communication`.

References

[ACF+97] J. Atserias, S. Climent, X. Farreres, G. Rigau, and H. Rodríguez. 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, 1997.

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

[AMS97] A. Artale, B. Magnini, and C. Strapparava. Lexical Discrimination with the Italian Version of WordNet. In *Proceedings of ACL Workshop Automatic Information Extraction and Building of Lexical Semantic Resources*, Madrid. Spain, 1997.

[BCE+98] L. Benítez, S. Cervell, G. Escudero, M. López, G. Rigau, and M. Taulé. Methods and Tools for Building the Catalan WordNet. In *Proceedings*

of ELRA Workshop on Language Resources for European Minority Languages, Granada, Spain, 1998.

[BG92] R. Bruce and L. Guthrie. Genus disambiguation: A study in weigthed preference. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING'92)*, Nantes, France, 1992.

[ER78] J. O. Eklundh and A. Rosenfeld. Convergence Properties of Relaxation Labelling. Technical Report 701, Computer Science Center. University of Maryland, 1978.

[FRR98] X. Farreres, G. Rigau, and H. Rodríguez. Using WordNet for Building WordNets. In *Proceedings of COLING-ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, Montréal, Canada, 1998.

[Gó98] A. Gómez. *J. Liebowitz (ed.). The Handbook of Applied Expert Systems.* CRC Press, 1998.

[Har83] R. M. Haralick. An interpretation for Probabilistic Relaxation. *Computer Vision, Graphics & Image Processing*, 22:388–395, 1983.

[HF97] B. Hamp and H. Feldweg. GermaNet - a Lexical-Semantic Net for German. In *Proceedings of ACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources*, Madrid. Spain, 1997.

[HZ83] R. A. Hummel and S. W. Zucker. On the foundations of relaxation labelling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(3), 1983.

[KF86] J. Kittler and J. Föglein. On Compatibility and Support Functions in Probabilistic Relaxation. *Computer Vision, Graphics & Image Processing*, 34:257–267, 1986.

[KI85] J. Kittler and J. Illingworth. Relaxation Labelling Algorithms - A Review. *Image & Vision Computing*, 3(4), 1985.

[KL94] K. Knight and S. Luk. Building a Large-Scale Knowledge Base for Machine Translation. In *Proceedings of the American Association for Artificial Inteligence (AAAI'94)*, 1994.

[Llo83] S. A. Lloyd. An optimization approach to relaxation labelling algorithms. *Image and Vision Computing*, 1(2):85–91, 1983.

[MP97] Lluís Màrquez and Lluís Padró. 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 1997.

[OH94] A. Okumura and E. Hovy. Building japanese-english dictionary based on ontology for machine translation. In *proceedings of ARPA Workshop on Human Language Technology*, pages 236–241, 1994.

[Pad96] Lluís Padró. POS Tagging Using Relaxation Labelling. In *Proceedings of the 16th International Conference on Computational Linguistics, COLING*, pages 877–882, Copenhagen, Denmark, August 1996.

[Pad98] Lluís Padró. *A Hybrid Environment for Syntax–Semantic Tagging*. Phd. Thesis, Dep. Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya, February 1998. http://www.lsi.upc.es/~padro.

[PBR+97] P. Vossen P, L. Bloksma, H. Rodríguez, S. Climent, A. Roventini, F. Bertagna, and A. Alonge. The EuroWordNet Base Concepts and Top-Ontology. Technical report, Deliverable D017D034D036 EuroWordNet LE2-4003, 1997.

[Pel79] S. Peleg. Monitoring Relaxation Labelling Algorithms Using Labelling Evaluation. Technical Report 842, Computer Vision Laboratory. Computer Science Center, 1979.

[PM94] M. Pelillo and A. Maffione. Using Simulated Annealing to Train Relaxation Labelling Processes. In *Proceedings of ICANN '94*, 1994.

[PR94] M. Pelillo and M. Refice. Learning Compatibility Coefficients for Relaxation Labeling Processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9), 1994.

[RAA97] German Rigau, Jordi Atserias, and Eneko Agirre. Combining Unsupervised Lexical Knowledge Methods for Word Sense Disambiguation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics. Joint ACL/EACL*, pages 48–55, Madrid, Spain, July 1997.

[RHZ76] R. Rosenfeld, R. Hummel, and S. Zucker. Scene labelling by relaxation operations. *IEEE Transactions on Systems, Man and Cybernetics*, 6(6):420–433, 1976.

[Ric97] S. Richardson. *Determining Similarity and Inferring Relations in a Lexical Knowledge Base*. Phd. Thesis, The City University of New York, New York, NY, 1997.

[RLS81] J. Richards, D. Landgrebe, and P. Swain. On the accuracy of pixel relaxation labelling. *IEEE Transactions on Systems, Man and Cybernetics*, 11(4):303–309, 1981.

[RRA98] G. Rigau, H. Rodríguez, and E. Agirre. Building Accurate Semantic Taxonomies from MRDs. In *Proceedings of COLING-ACL'98*, Montréal, Canada, 1998.

[RRT95] G. Rigau, H. Rodríguez, and J. Turmo. Automatically extracting Translation Links using a wide coverage semantic taxonomy. In *proceedings 15th International Conference AI'95*, Montpellier, France, 1995.

[Tor89] C. Torras. Relaxation and Neural Learning: Points of Convergence and Divergence. *Journal of Parallel and Distributed Computing*, 6:217–244, 1989.

[UH97] M. Utiyama and K. Hasida. Bottom-up Alignment of Ontologies. In *Proceedings of IJCAI workshop on Ontologies and Multilingual NLP*, Nagoya, Japan, 1997.

[VP97] Atro Voutilainen and Lluís Padró. Developing a Hybrid NP Parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing, ANLP*, pages 80–87, Washington DC, 1997. ACL.

[ZKH78] S. W. Zucker, E. V. Krishnamurty, and R. L. Haar. Relaxation processes for scene labelling: Convergence, speed and stability. *IEEE Transactions on Systems, Man and Cybernetics*, 8(1), 1978.

[ZLM81] S. W. Zucker, Y. G. Leclerc, and J. L. Mohammed. Continuous Relaxation and local maxima selection: Conditions for equivalence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(2), 1981.