

Enriching a Lexical Semantic Net with Selectional Preferences by Means of Statistical Corpus Analysis

Andreas Wagner¹

Abstract. Broad-coverage ontologies which represent lexical semantic knowledge are being built for more and more natural languages. Such resources provide very useful information for word sense disambiguation, which is crucial for a variety of NLP tasks (e.g. semantic annotation of corpora, information retrieval, or semantic inferencing). Since the manual encoding of such ontologies is very labour-intensive, the development of (semi-)automatic methods for acquiring lexical semantic information is an important task. This paper addresses the automatic acquisition of selectional preferences of verbs by means of statistical corpus analysis. Knowledge about such preferences is essential for inducing thematic relations, which link verbal concepts to nominal concepts that are selectionally preferred as their complements. Several approaches for learning selectional preferences from corpora have been proposed in the last years. However, their usefulness for ontology building is limited. This paper introduces a modification of one of these methods (i.e. the approach of Li & Abe [1]) and evaluates it by employing a gold standard. The results show that the modified approach is much more appropriate for the given task.

1 INTRODUCTION

Recently, broad-coverage, general-purpose lexical semantic ontologies have become available and/or are being developed for a variety of natural languages, e.g. WordNet [4] and EuroWordNet [12]. WordNet was developed at Princeton University for English and is widely used in the NLP community. EuroWordNet is a multilingual lexical semantic database which comprises WordNet-like ontologies for eight European languages. These wordnets are connected to an interlingual index so that a node in one language-specific wordnet can be mapped to the corresponding node in another language-specific wordnet. These resources capture the semantic properties of the most common words in a language. In particular, they encode the different senses of words (represented by the concept nodes of the ontology) and the basic semantic relations between word senses like hyponymy, antonymy, etc. (represented by the edges of the ontology).² Such resources contain useful information for word sense disambiguation, which is a prerequisite for several NLP tasks like semantic annotation of corpora, text analysis, information retrieval, or semantic inferencing. Thus, the resources provide necessary information for various kinds of NLP tools. Their intention is to capture general, domain-independent knowledge which complements domain-specific knowledge needed for a particular NLP system.

The most important semantic relation in the above-mentioned ontologies is hyponymy/hyperonymy. This relation constitutes a hierarchical structuring of the different semantic concepts. In WordNet, for example, the concept <life form>³ is a hyperonym of concepts like <animal>, <human>, and <plant>. Other semantic relations are in general not encoded exhaustively (or even not at all). However, they also provide useful information for NLP tasks. One group of such relations in EuroWordNet are thematic role relations. These relations connect verbal concepts with nominal concepts which typically occur as their complements. For example, the verbal concept <eat> should have AGENT pointers to the nominal concepts <human> and <animal>, and a PATIENT pointer to the concept <food>. Thematic relations provide information about the selectional preferences which verbs impose on their complements. This kind of information is useful for lexical and syntactic disambiguation (cf. [8], [1]).

As manual encoding of ontologies is very labour-intensive, (semi-)automatic methods have been explored, particularly the extraction of information from other existing lexical resources. However, such resources are often not complete or not available at all. For example, thematic relations are encoded in several language-specific wordnets in EuroWordNet, but only for some minor portions of verb concepts so that a mapping of these relations to another language does not yield an exhaustive coverage.

If appropriate lexical resources are missing, other means of automatically acquiring lexical information have to be considered. One possibility is the statistical analysis of corpora. This paper addresses the usefulness of employing statistical methods for learning thematic relations. In particular, I will investigate the acquisition of selectional preferences that verbs impose on their complements. Knowledge about selectional preferences is a prerequisite for encoding thematic relations.

Several approaches for the statistical acquisition of selectional preferences (represented as WordNet noun classes) have been proposed ([9], [10], [1]). As these approaches investigate corpora, i.e. huge collections of sentences, they reveal preferences for *syntactic* arguments. As thematic roles can have different syntactic realizations, the preferences for syntactic complements have to be mapped to the corresponding roles. This can be done manually or (semi-)automatically, cf. [6] and [7] for some basic approaches to solve this problem.

If selectional preferences are gathered to supplement an ontology, it is desirable (if not necessary) to find a representation for them which is both *empirically adequate* (i.e. captures all and only the preferred concepts) and *as compact as possible*. For example, it is not desirable to introduce PATIENT relations from <eat> to all the food

¹ Seminar für Sprachwissenschaft, University of Tübingen, Wilhelmstr. 113, D-72074 Tübingen, Germany, email: wagner@sfs.nphil.uni-tuebingen.de

² The synonymy relation is encoded within the nodes, since a concept is represented by a set of synonyms.

³ In this paper, I will enclose concepts in pointed brackets and use capital letters for relation identifiers.

concepts in the wordnet (<meat>, <strawberry cake>, ...) because this would be highly redundant and would not express any generalization. One would rather want to find a class which subsumes all the preferred classes (such as <food>). Of course, a class which is so general that it also subsumes dispreferred classes is unacceptable as well (e.g. <entity>). Thus, the problem is to find the *appropriate level* of generalization. The “compactness desideratum” (find classes which are not too specific) is particularly important for our task, the extension of a semantic net. It is motivated from a practical point of view (storage economy) as well as by conceptual considerations (appropriate generalizations should be expressed; this is important for applications like semantic inferencing).

This paper is organized as follows: Section 2 examines the suitability of the above-mentioned statistical methods for finding the appropriate generalization level. I will describe in detail the Li & Abe approach [1], which is explicitly intended for that task. I will report an experiment which reveals an inherent problem of this approach with respect to generalization. In section 3, I will introduce a modification of the method to overcome this problem, which is indicated by an analogous experiment. Section 4 describes a more systematic evaluation of the alternative approaches against a gold standard which I extracted from the EuroWordNet database. Section 5 gives a conclusion and sketches future work.

2 ACQUIRING SELECTIONAL PREFERENCES FROM CORPORA

Among the approaches for the acquisition of selectional preferences mentioned above, only the work of Li & Abe systematically addresses the problem of appropriate generalization. Resnik [9] does not determine a set of classes that represents selectional preferences at all. Ribas [10] determines such a set by a simple greedy algorithm. The impact of this algorithm for the generalization level of the selected classes is undetermined. Li & Abe obtain a set of classes that form a partition of the corpus instances. They employ a theoretically well-founded principle (Minimum Description Length) to find the appropriate generalization level.

In this section, I describe this method and the experiment I carried out to test its behaviour.

2.1 Information theoretic foundations

Information theory deals with coding information as efficiently as possible. In the framework of this discipline, information is usually coded in bits. If one has to code a sequence of signs (in our case, nouns which occur as the complement of a certain verb in a corpus), the simplest way to do this would be to represent each sign by a bit sequence of uniform length. However, if the probabilities of the individual signs differ significantly, it is more efficient (with respect to data compression) to assign shorter bit sequences to more probable (and thus more frequent) signs and longer bit sequences to less probable (and less frequent) signs. It can be shown that one can achieve the shortest average code length by assigning $\lceil \log_2 \frac{1}{p(x)} \rceil$ bits to a sign x with probability $p(x)$ (cf. [3]). Thus, if one has a good estimation of the probability distribution which underlies the occurrence of the signs, one can develop an efficient coding scheme (a mapping between signs and bit sequences) based on this estimation.

2.2 The basic method

The approach in [1] is based on the *Minimum Description Length (MDL) Principle* invented by J. Rissanen (cf. [11]). This principle

depends on the assumption that learning can be seen as data compression. The better one knows which general principles underlie a given data sample, the better one can make use of them to encode this sample efficiently. If one wants to encode a sample, one has to encode (a) the probability model that determines a coding scheme, and (b) the data themselves (by employing that coding scheme). The MDL principle states that the best probability model is that which achieves the highest data compression, i.e. which minimizes the sum of the lengths of (a) and (b). (The length of (a) is called *model description length*, the length of (b) *data description length*.) In our case, a sample consists of the noun tokens that appear at a certain syntactic argument slot (e.g. the direct object of a certain verb in the examined corpus).

Li & Abe represent the selectional behaviour of a verb (with respect to a certain argument) as a so-called *tree cut model*. Such a model provides a horizontal cut through the noun hierarchy tree, so that the classes that are located along the cut form a partition of the noun senses covered by the hierarchy. Each class is assigned a preference value. The preference value for a class in the cut is inherited by its subclasses. A tree cut model (cut + preference values) determines a probability distribution over the sample (see below), and hence a coding scheme. (Examples of tree cut models can be found in Tables 1–6.)

As preference value, Li & Abe estimate the so-called *association norm*:

$$A(c, v) = \frac{p(c, v)}{p(c)p(v)} = \frac{p(c|v)}{p(c)} \quad (1)$$

This measure quantifies the ratio of the occurrence probability of a noun class c at a certain argument slot⁴ of a certain verb v and the expected occurrence probability of c at this slot if independence between c and v is assumed. This is equivalent to the ratio of the conditional probability of c given v and the probability of c regardless of a particular verb. An *association norm* greater than 1 indicates that v prefers c , an *association norm* smaller than 1 indicates that v disprefers c .

Given the marginal probabilities $p(n)$ of the noun senses (regardless of a particular verb),⁵ a tree cut model determines a probability distribution over the noun senses n in a sample. This follows from the constraint that the association norm of a class is inherited by its descendants. Every noun sense n is represented in the cut by exactly one class c_n . So we have

$$p(n|v) = A(n, v)p(n) = A(c_n, v)p(n) \quad (2)$$

MDL is used to get the tree cut model with the appropriate generalization level. The number of bits required to encode a sample S using a probability model M is given by

$$L(M) = L_{mod}(M) + L_{dat}(M) \quad (3)$$

with

$$L_{mod}(M) = L_{cut}(M) + L_{par}(M) \quad (4)$$

$L_{mod}(M)$ is the model description length, $L_{dat}(M)$ the data description length, $L_{cut}(M)$ is the code length needed to identify the cut through the hierarchy, and $L_{par}(M)$ is the code length needed to encode the parameters of the model (the association norms of the classes on the cut). Following the MDL principle, we search for the model M that minimizes $L(M)$.

⁴ This slot is not explicitly referred to in the formula.

⁵ These probabilities are also estimated on the basis of a tree cut model by employing the MDL principle; cf. [1] and [5] for details.

For simplicity, it is assumed that all possible cuts have uniform probability. Thus, L_{cut} is constant for all cuts. As we aim at minimizing the description length, we can neglect this term.

Li & Abe calculate $L_{par}(M)$ as

$$L_{par}(M) = K \left(\frac{\log |S|}{2} \right) \quad (5)$$

K is the number of parameters in M (i.e. the number of classes on the cut) and $|S|$ is the sample size. For every class on the cut, the association norm is represented by $\frac{\log |S|}{2}$ bits. This precision minimizes $L(M)$ for a given M (cf. [11]).⁶

The data description length is given by

$$L_{dat}(M) = - \sum_{n \in S} \log p_M(n|v, s) \quad (7)$$

where p_M is a probability distribution determined by M (cf. section 2.1).

If the tree cut is located near the root, then the model description length will be low because the model contains only few classes. However, the data description length will be high because the code for the data is based on the probability distribution of the classes in the model, not on the real probability distribution of the individual nouns. The greater the difference between the supposed distribution and the real one, the longer the code. And the coarser the classification is, the more the corresponding distribution p_M deviates from the real distribution. On the other hand, if the tree cut is located near the leaves, the reverse is true: the fine-grained classification fits the data well, resulting in a low data description length, but the great amount of classes increases the model description length. Minimizing the sum of these two description lengths yields a balance between compactness (expressing generalizations) and accuracy (fitting the data) of the model.

2.3 Implementational details

In essence, I used the algorithm described in [1] to obtain the tree cut model. However, some modifications were necessary or useful for practical reasons.

Firstly, some WordNet specific problems had to be solved. The algorithm requires that the class hierarchy is a tree where the leaves represent the word senses and the inner nodes represent semantic classes. However, WordNet is not a pure tree, but a DAG, and all nodes represent both word senses and semantic classes (e.g. the node <person#individual#someone> represents at the same time a semantic class and a particular word sense for the nouns “person”, “individual”, and “someone”. No hyponym of the class represents this sense. To handle this problem, I introduced for every inner node an additional node that captures the noun sense that the node represents and made this additional node a hyponym of the inner node. So

⁶ I actually introduced an optimization: It does not make sense to represent the parameter value 0 with $\frac{\log |S|}{2}$ bits. (An association norm is 0 if and only if the corresponding class has no instances in the sample.) A more efficient coding strategy is to mark the classes that have a non-zero parameter and represent the parameter values for those classes only. First you need K bits, one for each class, which indicate whether a class occurs in the sample or not. Then you need $\frac{\log |S|}{2}$ bits for every class that occurs in the sample. Thus,

$$L_{par}(M) = K + K_S \left(\frac{\log |S|}{2} \right) \quad (6)$$

K_S is the number of classes that have instances in the sample. With this modification, one saves $(K - K_S) \frac{\log |S|}{2} - K$ bits.

all word senses are represented by leaves. (These additional nodes will be indicated by ‘REST::’ as they represent the “rest” of class instances which is not captured by the subclasses.) To handle the DAG issue, I “broke the DAG into a tree”. This means, if a node has more than one parent, I virtually duplicated that node (and its descendants) to maintain a tree structure. This solution has the disadvantage that parts of the sample are artificially duplicated. I will work on a more principled solution in the future.

To eliminate noise, I introduced a threshold in the following way: The algorithm compares possible cuts by traversing the hierarchy top down. If a class with a probability smaller than 0.05 is encountered, then the traversal stops, i.e. the descendants of that class are not examined. This has also the advantage of limiting the search space.

2.4 Experiment

2.4.1 Setting

To test the behaviour of the Li & Abe approach with respect to generalization, I applied it to acquire selectional preferences for the direct object slot.⁷ I extracted verb–object instances from a portion of the British National Corpus (parts A–E; about 40 million words) with Steven Abney’s CASS parser [2].⁸ This resulted in a sample of about 2 million verb–noun pairs. Then I applied the algorithm of Li & Abe to calculate the selectional preferences of 24 test verbs and manually inspected the results.

2.4.2 Results

The experiment revealed a significant drawback of employing the MDL principle for our task. It turned out that the frequency of the examined verb in the sample has an undesirable impact on the generalization level of the tree cut model: The algorithm tends to over-generalize (acquire a tree cut with few general classes) for infrequent verbs and to under-generalize (acquire a tree cut with many specific classes) for frequent verbs. This behaviour is an immediate consequence of the MDL principle: If a large amount of data has to be described, then the model cost L_{mod} does not contribute much to the whole description length L . The gain of a complex model for encoding the data outweighs the model cost. If, however, only few data have to be described, then L_{mod} is much more significant for L : the cost of encoding a complex model outweighs the gain for encoding the data.

However, this is not the desired behaviour. Generalization should not be triggered by the sample size, but by the “semantic variety” of the instances in the sample: Nouns like “apple”, “pear”, “strawberry” should generalize to <fruit>. Further instances like “pork” or “cake” should trigger generalization to <food>, and yet further instances like “house” or “vessel” to <physical object>.

To illustrate these considerations, let us look at the verbs “kill”, “murder”, and “assassinate”. Tables 1–3 show (parts of) the tree cut models obtained for these verbs. For the rather frequent verb “kill” (3352 occurrences), hyponyms of <animal> are acquired. These classes are too specific; one would expect the class <life form>. In contrast, the tree cut model for the less frequent verb “murder” (477 occurrences) is an over-generalization. This verb prefers the

⁷ I chose this slot for several practical reasons. Of course, one can apply the algorithm to samples of other syntactic complements as well.

⁸ I thank Steven Abney and Marc Light, who made their source code for acquiring selectional preferences available to me. Those parts of my implementation which deal with collecting and storing co-occurrence statistics from the sample are adapted from their code.

more specific concept <person>. The over-generalization is even worse for the infrequent verb “assassinate” (79 occurrences). The selectional preference of this verb is even more specific; it prefers a concept like “important person” (which does not exist in WordNet). However, one of the most general concepts, <entity>, is retrieved.

Table 1. Part of the tree cut model for “kill” (standard MDL)

<i>class</i>	<i>A(class, verb)</i>
...	...
<person#individual#someone#mortal#human>	3.11
...	...
<herbivore>	31.42
<aquatic_vertibrate>	10.93
<bird>	9.02
<amphibian>	11.36
<reptile#reptilian>	5.44
<metatherian>	6.08
<livestock#farm_animal>	31.42
<bull>	109.96
<insectivore>	9.43
<aquatic_mammal>	88.81
<carnivore>	15.56
<lagomorph#gnawing_mammal>	178.59
<rodent#gnawer#gnawing_animal>	11.28
<ungulate#hoofed_mammal>	11.95
<primate>	150.40
<proboscidean#proboscidian>	20.95
<invertebrate>	18.21
<predator#predatory_animal>	31.42
<prey#quarry>	364.85
<REST::animal#animate_being#beast#brute>	343.82
<plant#flora#plant_life>	0.14
...	...
<REST::life_form#organism#being#living_thing>	19.47
...	...

Table 2. Part of the tree cut model for “murder” (standard MDL)

<i>class</i>	<i>A(class, verb)</i>
...	...
<life_form#organism#being#living_thing>	4.15
...	...

Table 3. Part of the tree cut model for “assassinate” (standard MDL)

<i>class</i>	<i>A(class, verb)</i>
...	...
<entity>	2.29
...	...

3 THE WEIGHTING ALGORITHM

3.1 Introducing a weighting factor

The problem discussed above is caused by different complexities of L_{par} and L_{dat} (with respect to the sample size $|S|$). As one can see from the equations (5) and (7), L_{par} has the complexity $O(\log |S|)$, while L_{dat} has the complexity $O(|S|)$. Thus, with growing $|S|$, L_{dat}

“grows faster” than L_{par} , and for frequent verbs, the model description length can be neglected, so that a model with many specific classes becomes “affordable”.

To overcome this drawback, I extended the expression to minimize by a weighting factor: Instead of minimizing $L_{par} + L_{dat}$, the modified algorithm minimizes

$$L_{par}(M) + C \left(\frac{\log |S|}{|S|} \right) L_{dat}(M) \quad (C > 0) \quad (8)$$

Now both addends have the same complexity. $|S|$ does not directly affect generalization any more.

The value of the constant C influences the degree of generalization. The smaller C is, the more general classes are acquired. The possibility of manipulating the overall generalization level by the choice of C introduces some flexibility which might prove useful when the algorithm is applied in different situations (tasks, domains, languages, etc.).

Note that the introduction of weighting is a deviation from the “pure” MDL principle that is based on the view that learning can be regarded as data compression. However, it can be shown that the modified algorithm is a kind of Bayesian learning.

3.2 Experiments

To test the impact of this modification on the generalization level of the acquired tree cuts, I examined verbs with diverse numbers of *different* noun complements (types) in the training sample. In particular, I selected all verbs with a high number (≥ 1000), a medium number (400–600), a low number (70–100), and a very low number (10–40) of different complements and compared the generalization level retrieved by the “standard MDL” algorithm and the weighting algorithm. (I arbitrarily chose $C = 50$.) For all verbs with a high number and 89% of the verbs with a medium number of complements, the weighting algorithm obtained more general classes than the standard MDL algorithm. In contrast, more specific classes were computed for almost all verbs with a low and a very low number of different complements (95.9% and 99.5%, respectively). Hence, the modification changes the behaviour of the algorithm towards the desired direction (that variety of complements should trigger generalization).

Tables 4–6 show the tree cut models for “kill”, “murder”, and “assassinate” which are yielded by the weighting algorithm ($C = 50$). Now these models are at the appropriate level of generalization.

Table 4. Part of the tree cut model for “kill” (weighting algorithm)

<i>class</i>	<i>A(class, verb)</i>
...	...
<life_form#organism#being#living_thing>	3.25
...	...

Table 5. Part of the tree cut model for “murder” (weighting algorithm)

<i>class</i>	<i>A(class, verb)</i>
...	...
<person#individual#someone#mortal#human>	4.65
...	...

Table 6. Part of the tree cut model for “assassinate” (weighting algorithm)

<i>class</i>	<i>A(class, verb)</i>
...	...
<adult>	4.67
<communicator>	4.74
<contestant>	9.40
<spiritual_leader>	14.81
<head#chief#top_dog>	74.25
<president#chairman#chairwoman#chair>	982.64
<REST::leader>	1187.54
<peer#equal#match#compeer>	24.04
<relative#relation>	6.47
<czar#tsar#tzar>	791.57
<king#male_monarch>	1187.36
<authority>	238.61
<suspect>	1187.36
...	...

4 EVALUATION

Up to now, it was not possible to automatically evaluate the “intuitiveness” of the selectional preferences acquired by a certain approach because there was no way to tell the computer which preferences correspond with human intuition. One only could manually inspect a few illustrative examples and concentrate on evaluating the performance of the approach in NLP tasks, e.g. word sense disambiguation (which is, of course, a crucial issue). The EuroWordNet database provides information suitable for compiling a gold standard. This gold standard allows to evaluate the lexicographic appropriateness (the appropriateness with respect to building wordnets) of an acquisition approach automatically and on a broader empirical basis. This section describes the evaluation of the standard MDL and the weighting algorithm.

4.1 Retrieving the gold standard

As mentioned in section 1, some of the wordnets in EuroWordNet contain thematic relations: the wordnets for Dutch, English, Estonian, Italian, and Spanish. These relations have been manually encoded or extracted from other lexical resources, respectively. I employed them for the gold standard by mapping them to WordNet (which does not contain thematic relations itself).

I started from the simplifying heuristic that the patient of a verb is usually syntactically realized as its direct object. In EuroWordNet, a verb sense is connected to a noun sense that it prefers as its patient by the INVOLVED_PATIENT relation. Thus, I mapped the relations of this type to WordNet.

I extracted those INVOLVED_PATIENT relations where both the source node and the target node were linked to a node in the inter-lingual index (ILI) by a synonymy or a near-synonymy relation.⁹ The inter-lingual index essentially consists of all the concept nodes of WordNet 1.5. Thus, extracting the ILI concepts equivalent to the source and the target concept of an INVOLVED_PATIENT relation, respectively, immediately yields a mapping of this relation to WordNet 1.5. With this procedure, I retrieved 605 relations.

However, a certain amount of these relations were inappropriate for our task. The assumption that a patient is syntactically realized as

⁹ Most concepts in the language-specific wordnets are linked to a corresponding concept in the ILI by a synonymy link. However, it is often the case that there is no ILI concept that exactly matches a language concept. This language concept has to be linked to a semantically related ILI concept, e.g. by a hyponymy or a hyperonymy link.

an object is a good starting point, but does not apply for all cases. Unaccusative verbs (e.g. <silt>) realize their patient (e.g. <sediment>) as their subject. Other verbs do not realize their patient as a syntactic argument at all (e.g. <delouse> – <louse>). Patients of such verbs cannot be found by examining verb objects.

Furthermore, some relations like <address> INVOLVED_PATIENT <addressee> indicate a noun concept that itself is perfectly adequate, but does not capture the majority of the noun instances in the corpus. Any noun referring to a human could occur as the patient of <address>. Thus, the learning algorithm should generalize to the <human> level. However, <addressee> is a subclass of <human> (which has no hyponyms itself). It makes sense to encode thematic relations where the noun concept does not subsume all preferred concepts extensionally, but characterizes them intensionally. However, such relations cannot be derived by generalizing from corpus instances. They could rather be acquired by examining derivational patterns.

To obtain a gold standard that is appropriate for the evaluation of the two algorithms, I excluded these problematic cases. 390 relations remained.

For every WordNet verb concept which was retrieved in this way, I collected all the verbs which the concept represents and assigned each of them the noun concepts linked to it. This means that the information to which sense of the verb a noun concept is related is lost. However, this is necessary to perform the comparison with the results of the two algorithms because they compute preferences for verbs, not verb senses. I obtained 599 verbs altogether (excluding multiword lexemes).

4.2 Evaluation results

The intersection of the verbs in the gold standard and the verbs in the training sample contained 522 verbs which were connected to 1082 noun concepts in the gold standard. For both algorithms (and different values of C), I evaluated the match between the classes acquired for a verb and the gold standard classes for that verb. Table 7 shows the number and the percentage of the noun classes in the gold standard which were exactly matched, not matched at all,¹⁰ or matched by more general or more specific classes in the tree cut model. This table contains recall values (number of correct classes that are found). Note that it does not make sense to calculate precision (number of preferred classes in the tree cut model that are correct) because the gold standard does not capture *every* sense of a verb, i.e. it is not “complete” with respect to a particular verb.

4.3 Discussion

The results show that the weighting algorithm significantly outperforms the standard MDL algorithm. For $C = 10,000$ or higher, about three times as many classes are matched than with standard MDL. The same holds true if we add the exact matches and the 1-level deviations, which are the most helpful cases in a scenario in which selectional preferences are acquired automatically and corrected manually afterwards (33% vs. 11%). The classes with the most exact matches ($C \geq 10,000$) are <food> (18 times), <physical object> (10 times), <animal> (9 times), <beverage> (7 times), and <person> (6 times).

The choice of C influences the performance. Here, selecting a high value (which forces specific tree cuts) improves the results, but only

¹⁰ Dispreferred classes, i.e. classes with a preference value < 1 , are considered as not matching the gold standard.

Table 7. Comparison of acquired tree cut models with the gold standard

number (percentage) of gold standard classes	standard MDL	weighting: $C =$			
		100	1,000	10,000	100,000
exactly matched	57 (5.3%)	106 (9.8%)	160 (14.8%)	162 (15.0%)	162 (15.0%)
matched by 1 level hyperonym	56 (5.2%)	90 (8.3%)	120 (11.1%)	126 (11.6%)	126 (11.6%)
matched by 1 level hyponym	5 (0.5%)	38 (3.5%)	63 (5.8%)	69 (6.4%)	69 (6.4%)
matched by 2 level hyperonym	107 (9.9%)	86 (7.9%)	122 (11.3%)	125 (11.6%)	125 (11.6%)
matched by 2 level hyponym	0 (0.0%)	6 (0.6%)	6 (0.6%)	6 (0.6%)	6 (0.6%)
matched by ≥ 3 level hyperonym	524 (48.4%)	296 (27.4%)	193 (17.8%)	194 (17.9%)	194 (17.9%)
matched by ≥ 3 level hyponym	0 (0.0%)	9 (0.8%)	11 (1.0%)	11 (1.0%)	11 (1.0%)
not matched	333 (30.8%)	451 (41.7%)	407 (37.6%)	389 (36.0%)	389 (36.0%)

to a certain extent. Above $C = 10,000$, no improvement can be observed. The tree cuts have reached their “lower limit” then. This limit is determined by the threshold introduced to eliminate noise (cf. section 2.3).

The percentage of classes which were not matched at all is higher for the weighting algorithm. The reason for this is that the majority of verbs occur rather infrequently (cf. Zipf’s law) so that standard MDL tends to acquire over-general classes for them. Thus, the chance that the class in the gold standard is subsumed by such a general class is higher. (Note that most of the classes matched by standard MDL are at least 3 levels too general.)

However, even with the weighting algorithm the overall results are not satisfying. 15% of the classes in the gold standard are exactly matched; 33% are approximated with 0 or 1 level deviation. 41.1% are matched by too general classes, but only 8% by too specific classes. More than one third of the classes is not found at all.

The main reason for this behaviour is that the selectional preferences are acquired for verb forms, not for verb senses. Calculating a tree cut model for a highly polysemous verb may trigger inappropriate generalizations, since the different senses of the verb could introduce a high variety of complement nouns, which yields generalization, even if each sense alone prefers rather specific noun concepts.

On the other hand, it would be useful to pool verb instances which represent the same concept when calculating tree cut models. For example, the verbs “arrest”, “nail”, “nab”, and “cop” are represented by the same concept in the gold standard. More appropriate selectional preferences could be acquired if the algorithm did not compute one tree cut for all instances of “arrest”, one for all instances of “nail” etc., but one tree cut for all instances of “arrest”, “nail”, “nab”, and “cop” which have the same sense. This would also reduce the percentage of unmatched classes, since verb instances which have a sense that does not occur in the gold standard would not be taken into account.

5 CONCLUSION AND FUTURE WORK

In this paper, I addressed the automatic acquisition of selectional preferences by the statistical analysis of corpora in order to be encoded in lexical semantic ontologies. I argued that methods which have been proposed for the acquisition of selectional preferences do not satisfyingly cope with the task of finding the appropriate generalization level. I modified one of these approaches and showed that the modified approach is much better suited for computing generalization levels which are appropriate for ontology building. The EuroWordNet database provides information that can be combined to

obtain a gold standard for selectional preferences. With this gold standard, lexicographic appropriateness can be evaluated automatically and on a broader empirical basis. This evaluation shows that the algorithm proposed in this paper is promising. However, the results are not satisfying yet. One shortcoming of the experiments described here (as well as in the mentioned work of Resnik, Ribas and Li & Abe) is that the learning algorithms are fed with word forms rather than word senses, which would be adequate. Employing corpora which are at least partially semantically disambiguated should improve the performance significantly.

In the near future, I will employ approaches for lexical disambiguation and test their impact on the performance of the weighting algorithm. Furthermore, I will test the methods described in this paper for different argument slots. As large syntactically annotated corpora are becoming more and more available, other verb–argument relations than direct objects can be reliably extracted and fed into the learning algorithm.

REFERENCES

- [1] Naoki Abe and Hang Li, ‘Learning Word Association Norms Using Tree Cut Pair Models’, in *Proc. of 13th Int. Conf. on Machine Learning*, (1996).
- [2] Steven Abney, ‘Partial parsing via finite-state cascades’, in *Workshop on Robust Parsing (ESSLLI ’96)*, ed., John Carroll, pp. 8 – 15, (1996).
- [3] Thomas M. Cover and Joy A. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, 1991.
- [4] *WordNet: An electronic lexical database*, ed., Christiane Fellbaum, MIT Press, Cambridge, Mass., 1998.
- [5] Hang Li and Naoki Abe, ‘Generalizing Case Frames Using a Thesaurus and the MDL Principle’, in *Proc. of Int. Conf. on Recent Advances in NLP*, (1995).
- [6] Diana McCarthy and Anna Korhonen. Detecting verbal participation in diathesis alternations, 1998. Proc. of 36th Annual Meeting of the Association for Computational Linguistics.
- [7] Wim Peters, ‘Corpus-based conceptual characterisation of verbal predicate structures’, in *Proc. of Computational Linguistics in the Netherlands*, Antwerpen, (1996).
- [8] Philip Resnik, ‘Selectional preference and sense disambiguation’, in *ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington, D.C., (1997).
- [9] Philip Stuart Resnik, *Selection and Information: A Class-Based Approach to Lexical Relationships*, Dissertation, University of Pennsylvania, 1993.
- [10] Francesc Ribas, ‘An experiment on learning appropriate selectional restrictions from a parsed corpus’, in *Proc. of COLING*, Kyoto, (1994).
- [11] Jorma Rissanen and Eric Sven Ristad, ‘Language acquisition in the MDL framework’, in *Language Computations*, ed., Eric Sven Ristad, volume 17 of *Series in Discrete Mathematics and Theoretical Computer Science*, 149–166, DIMACS, (1992).
- [12] Piek Vossen, ed. *EuroWordNet Final Document*. EuroWordNet (LE2-4003, LE4-8328), 1999. Deliverable D032D033/2D014.