Tree-cut and A Lexicon based on Systematic Polysemy

Noriko Tomuro
DePaul University
School of Computer Science, Telecommunications and Information Systems
243 S. Wabash Ave.
Chicago, IL 60604
tomuro@cs.depaul.edu

Abstract

This paper describes a lexicon organized around *systematic polysemy*: a set of word senses that are related in systematic and predictable ways. The lexicon is derived by a fully automatic extraction method which utilizes a clustering technique called *tree-cut*. We compare our lexicon to WordNet *cousins*, and the inter-annotator disagreement observed between WordNet Semcor and DSO corpora.

1 Introduction

In recent years, the granularity of word senses for computational lexicons has been discussed frequently in Lexical Semantics (for example, (Kilgarriff, 1998a; Palmer, 1998)). This issue emerged as a prominent problem after previous studies and exercises in Word Sense Disambiguation (WSD) reported that, when fine-grained sense definitions such as those in WordNet (Miller, 1990) were used, entries became very similar and indistinguishable to human annotators, thereby causing disagreement on correct tags (Kilgarriff, 1998b; Veronis, 1998; Ng *et al.*, 1999). In addition to WSD, the selection of sense inventories is fundamentally critical in other Natural Language Processing (NLP) tasks such as Information Extraction (IE) and Machine Translation (MT), as well as in Information Retrieval (IR), since the difference in the correct sense assignments affects recall, precision and other evaluation measures.

In response to this, several approaches have been proposed which group fine-grained word senses in various ways to derive coarse-grained sense groups. Some approaches utilize an abstraction hierarchy defined in a dictionary (Kilgarriff, 1998b), while others utilize surface syntactic patterns of the functional structures (such as predicate-argument structure for verbs) of words (Palmer, 1998). Also, the current version of WordNet (1.6) encodes groupings of similar/related word senses (or synsets) by a relation called *cousin*.

Another approach to grouping word senses is to utilize a linguistic phenomenon called *systematic polysemy*: a set of word senses that are related in systematic and predictable ways.[1] For example, ANIMAL and MEAT meanings of the word "chicken" are related because chicken as meat refers to the flesh of a chicken as a bird that is used for food.[2] This relation is systematic, since many ANIMAL words such as "duck" and "lamb" have a MEAT meaning. Another example is the relation QUANTITY-PROCESS observed in nouns such as "increase" and "supply".

Sense grouping based on systematic polysemy is lexico-semantically motivated in that it expresses general human knowledge about the *relatedness* of word meanings. Such sense groupings have advantages compared to other approaches. First, related senses of a word often exist simultaneously in a discourse (for example the QUANTITY and PROCESS meanings of "increase" above). Thus, systematic polysemy can be effectively used in WSD (and WSD evaluation) to accept multiple or alternative sense tags (Buitelaar, personal communication). Second, many systematic relations are observed between senses which belong to different semantic categories. So if a lexicon is defined by a collection of separate trees/hierarchies (such as the case of WordNet), systematic polysemy can express similarity between senses that are not hierarchically proximate. Third, by explicitly representing (inter-)relations between senses, a lexicon based on systematic polysemy can facilitate semantic inferences. Thus it is useful in knowledge-intensive NLP tasks such as discourse analysis, IE and MT. More recently, (Gonzalo *et al.*, 2000) also discusses potential usefulness of systematic polysemy for clustering word senses for IR.

However, extracting systematic relations from large sense inventories is a difficult task. Most often, this procedure is done manually. For example, WordNet cousin relations were identified manually by the WordNet lexicographers. A similar effort was also made in the EuroWordnet project (Vossen *et*

[1] Systematic polysemy (in the sense we use in this paper) is also referred to as *regular polysemy* (Apresjan, 1973) or *logical polysemy* (Pustejovsky, 1995).

[2] Note that systematic polysemy should be contrasted with *homonymy*, which refers to words which have more than one unrelated sense (e.g. FINANCIAL_INSTITUTION and SLOPING_LAND meanings of the word "bank").

al., 1999). The problem is not only that manual inspection of a large, complex lexicon is very time-consuming, it is also prone to inconsistencies.

In this paper, we describes a lexicon organized around systematic polysemy. The lexicon is derived by a fully automatic extraction method which utilizes a clustering technique called *tree-cut* (Li and Abe, 1998). In our previous work (Tomuro, 2000), we applied this method to a small subset of WordNet nouns and showed potential applicability. In the current work, we applied the method to all nouns and verbs in WordNet, and built a lexicon in which word senses are partitioned by systematic polysemy. We report results of comparing our lexicon with the WordNet cousins as well as the inter-annotator disagreement observed between two semantically annotated corpora: WordNet Semcor (Landes *et al.*, 1998) and DSO (Ng and Lee, 1996). The results are quite promising: our extraction method discovered 89% of the WordNet cousins, and the sense partitions in our lexicon yielded better κ values (Carletta, 1996) than arbitrary sense groupings on the agreement data.

2 The Tree-cut Technique

The tree-cut technique is an unsupervised learning technique which partitions data items organized in a tree structure into mutually-disjoint clusters. It was originally proposed in (Li and Abe, 1998), and then adopted in our previous method for automatically extracting systematic polysemy (Tomuro, 2000). In this section, we give a brief summary of this tree-cut technique using examples from (Li and Abe, 1998)'s original work.

2.1 Tree-cut Models

The tree-cut technique is applied to data items that are organized in a structure called a *thesaurus tree*. A thesaurus tree is a hierarchically organized lexicon where leaf nodes encode lexical data (i.e., words) and internal nodes represent abstract semantic classes. A *tree-cut* is a partition of a thesaurus tree. It is a list of internal/leaf nodes in the tree, and each node represents a set of all leaf nodes in a subtree rooted by the node. Such a set is also considered as a *cluster*.[3] Clusters in a tree-cut exhaustively cover all leaf nodes of the tree, and they are mutually disjoint. For instance, Figure 1 shows an example thesaurus tree and one possible tree-cut [AIRCRAFT, ball, kite, puzzle], which is indicated by a thick curve in the figure. There are also four other possible tree-cuts for this tree: [airplane, helicopter, ball, kite, puzzle], [airplane, helicopter, TOY], [AIRCRAFT, TOY] and [ARTIFACT].

In (Li and Abe, 1998), the tree-cut technique was applied to the problem of acquiring generalized case frame patterns from a corpus. Thus, each node/word in the tree received as its value the number of instances where the word occurred as a case role (subject, object etc.) of a given verb. Then the acquisition of a generalized case frame was viewed as a problem of selecting the best tree-cut model that estimates the true probability distribution, given a sample corpus data.

Formally, a *tree-cut model* M is a pair consisting of a tree-cut Γ and a probability parameter vector Θ of the same length,

$$M = (\Gamma, \Theta) \tag{1}$$

where Γ and Θ are:

$$\Gamma = [C_1, .., C_k], \Theta = [P(C_1), .., P(C_k)] \tag{2}$$

where C_i $(1 \leq i \leq k)$ is a cluster in the tree-cut, $P(C_i)$ is the probability of a cluster C_i, and $\sum_{i=1}^{k} P(C_i) = 1$. Note that $P(C)$ is the probability of cluster $C = \{n_1, .., n_m\}$ as a whole, that is, $P(C) = \sum_{j=1}^{m} P(n_j)$. For example, suppose a corpus contains 10 instances of verb-object relation for the verb "fly", and the frequencies of object nouns n, denoted $f(n)$, are as follows: $f(airplane) = 5, f(helicopter) = 3, f(ball) = 0, f(kite) = 2, f(puzzle) = 0$. Then, the set of tree-cut models for the example thesaurus tree shown in Figure 1 includes ([airplane, helicopter, TOY], [.5, .3, .2]) and ([AIRCRAFT, TOY], [.8, .2]).

2.2 The MDL Principle

To select the best tree-cut model, (Li and Abe, 1998) uses the *Minimal Description Length (MDL)*. The MDL is a principle of data compression in Information Theory which states that, for a given dataset, the best model is the one which requires the minimum length (often measured in bits) to encode the model (the *model description length*) and the data (the *data description length*) (Rissanen, 1978). Thus, the MDL principle captures the trade-off between the simplicity of a model, which is measured by the number of clusters in a tree-cut, and the goodness of fit to the data, which is measured by the estimation accuracy of the probability distribution.

The calculation of the description length for a tree-cut model is as follows. Given a thesaurus tree T and a sample S consisting of the case frame instances, the total description length $L(M, S)$ for a tree-cut model $M = (\Gamma, \Theta)$ is

$$L(M, S) = L(\Gamma) + L(\Theta|\Gamma) + L(S|\Gamma, \Theta) \tag{3}$$

where $L(\Gamma)$ is the model description length, $L(\Theta|\Gamma)$ is the parameter description length (explained shortly), and $L(S|\Gamma, \Theta)$ is the data description length. Note that $L(\Gamma) + L(\Theta|\Gamma)$ essentially corresponds to the usual notion of the model description length.

[3] A leaf node is also a cluster whose cardinality is 1.

Γ	L(Θ|Γ)	L(S|Γ,Θ)	L(M,S)
[A]	1.66	11.60	13.26
[AC,TOY]	3.32	14.34	17.66
[ap,heli,TOY]	4.98	14.44	19.42
[AC,ball,kite,puz]	6.64	4.96	11.60
[ap,hel,ball,kite,puz]	8.31	5.06	13.37

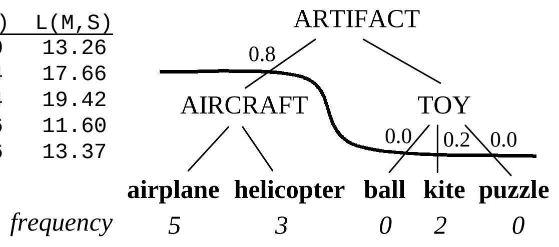

Figure 1: The MDL lengths and the final tree-cut

Each length in $L(M, S)$ is calculated as follows.[4] The model description length $L(\Gamma)$ is

$$L(\Gamma) = log_2|G| \tag{4}$$

where G is the set of all cuts in T, and $|G|$ denotes the size of G. This value is a constant for all models, thus it is omitted in the calculation of the total length.

The parameter description length $L(\Theta|\Gamma)$ indicates the complexity of the model. It is the length required to encode the probability distribution of the clusters in the tree-cut Γ. It is calculated as

$$L(\Theta|\Gamma) = \frac{k}{2} \times log_2|S| \tag{5}$$

where k is the length of Θ, and $|S|$ is the size of S.

Finally, the data description length $L(S|\Gamma, \Theta)$ is the length required to encode the whole sample data. It is calculated as

$$L(S|\Gamma, \Theta) = -\sum_{n \in S} log_2 P(n) \tag{6}$$

where, for each $n \in C$ and each $C \in \Gamma$,

$$P(n) = \frac{P(C)}{|C|} \text{ and } P(C) = \frac{f(C)}{|S|} \tag{7}$$

Note the equation (7) essentially computes the *Maximum Likelihood Estimate (MLE)* for all n.[5]

A table in Figure 1 shows the MDL lengths for all five tree-cut models. The best model is the one with the tree-cut [AIRCRAFT, ball, kite, puzzle].

3 Clustering Systematic Polysemy

Using the tree-cut technique described above, our previous work (Tomuro, 2000) extracted systematic polysemy from WordNet. In this section, we give a summary of this method, and describe the cluster pairs obtained by the method.

3.1 Extraction Method

In our previous work, systematically related word senses are derived as binary cluster pairs, by applying the extraction procedure to a combination of two WordNet (sub)trees. This process is done in the following three steps. In the first step, all leaf nodes of the two trees are assigned a value of either 1, if a node/word appears in both trees, or 0 otherwise.[6] In the second step, the tree-cut technique is applied to each tree separately, and two tree-cuts (or sets of clusters) are obtained. To search the best tree-cut for a tree (i.e., the model which requires the minimum total description length), a greedy algorithm called *Find-MDL* described in (Li and Abe, 1998) is used to speed up the search. Finally in the third step, clusters in those two tree-cuts are matched up, and the pairs which have substantial overlap (more than three overlapping words) are selected as systematic polysemies.

Figure 2 shows parts of the final tree-cuts for the `ARTIFACT` and `MEASURE` classes. Note in the figure, bold letters indicate words which are polysemous in the two trees (i.e., assigned a value 1).

3.2 Modification

In the current work, we made a minor modification to the extraction method described above, by removing nodes that are assigned a value 0 from the trees. The purpose was to make the tree-cut technique less sensitive to the structure of a tree and produce more specific clusters defined at deeper levels.[7] The MDL principle inherently penalizes a complex tree-cut by assigning a long parameter length. Therefore, shorter tree-cuts partitioned at abstract levels are often preferred. This causes a problem when the tree is bushy, which is the case with WordNet trees. Indeed, many tree-cut clusters obtained in our previous work were from nodes at depth 1 (counting the root as depth 0) – around 88% (122

[4] For justification and detailed explanation of these formulas, see (Li and Abe, 1998).

[5] In our previous work, we used entropy instead of MLE. That is because the lexicon represents true population, not samples; thus there is no additional data to estimate.

[6] Prior to this, each WordNet (sub)tree is transformed into a thesaurus tree, since WordNet tree is a graph rather than a tree, and internal nodes as well as leaf nodes carry data. In the transformation, all internal nodes in a WordNet tree are copied as leaf nodes, and shared subtrees are duplicated.

[7] Removing nodes with 0 is also warranted since we are not estimating values for those nodes (as explained in footnote 5).

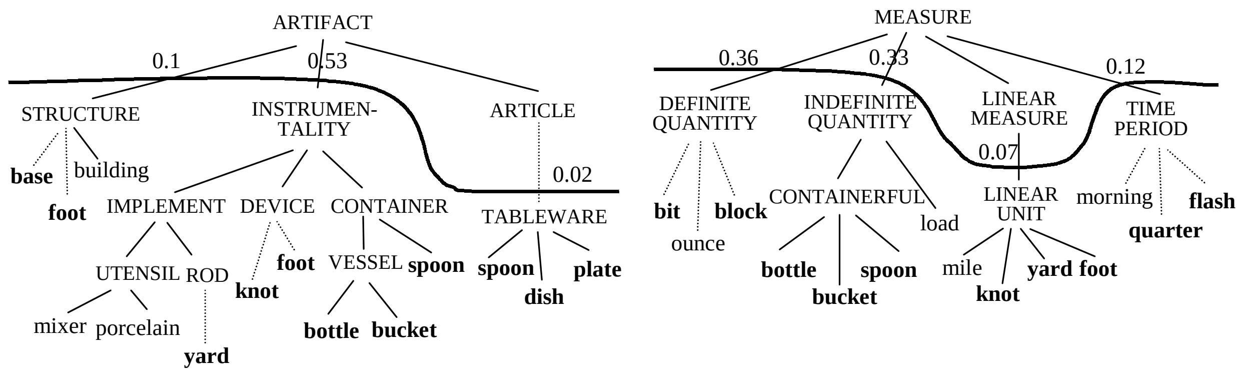

Figure 2: Parts of the final tree-cuts for ARTIFACT and MEASURE

Table 1: Automatically Extracted Cluster Pairs

Category	Basic classes	Underspecified classes	Cluster pairs
Nouns	24	99	2,377
Verbs	10	59	1,710
Total	34	158	4,077

out of total 138 clusters) obtained for 5 combinations of WordNet noun trees. Note that we did not allow a cluster at the root of a tree; thus, depth 1 is the highest level for any cluster. After the modification above, the proportion of depth 1 clusters decreased to 49% (169 out of total 343 clusters) for the same tree combinations.

3.3 Extracted Cluster Pairs

We applied the modified method described above to all nouns and verbs in WordNet. We first partitioned words in the two categories into *basic classes*. A basic class is an abstract semantic concept, and it corresponds to a (sub)tree in the WordNet hierarchies. We chose 24 basic classes for nouns and 10 basic classes for verbs, from WordNet Top categories for nouns and lexicographers' file names for verbs respectively. Those basic classes exhaustively cover all words in the two categories encoded in WordNet. For example, basic classes for nouns include ARTIFACT, SUBSTANCE and LOCATION, while basic classes for verbs include CHANGE, MOTION and STATE.

For each part-of-speech category, we applied our extraction method to all combinations of two basic classes. Here, a combined class, for instance ARTIFACT-SUBSTANCE, represents an *underspecified* semantic class. We obtained 2,377 cluster pairs in 99 underspecified classes for nouns, and 1,710 cluster pairs in 59 underspecified classes for verbs. Table 1 shows a summary of the number of basic and underspecified classes and cluster pairs extracted by our method.

Although the results vary among category combinations, the accuracy (precision) of the derived cluster pairs was rather low: 50 to 60% on average, based on our manual inspection using around 5% randomly chosen samples.[8] This means our automatic method over-generates possible relations. We speculate that this is because in general, there are many *homonymous* relations that are 'systematic' in the English language. For example, in the ARTIFACT-GROUP class, a pair [LUMBER, SOCIAL_GROUP] was extracted. Words which are common in the two clusters are "picket", "board" and "stock". Since there are enough number of such words (for our purpose), our automatic method could not differentiate them from true systematic polysemy.

4 Evaluation: Comparison with WordNet Cousins

To test our automatic extraction method, we compared the cluster pairs derived by our method to WordNet cousins. The cousin relation is relatively new in WordNet, and the coverage is still incomplete. Currently a total of 194 unique relations are encoded. A cousin relation in WordNet is defined between two synsets, and it indicates that senses of a word that appear in both of the (sub)trees rooted by those synsets are related.[9] The cousins were man-

[8] Note that the relatedness between clusters was determined solely by our subjective judgement. That is because there is no existing large-scale lexicon which encodes related senses completely for all words in the lexicon. (Note that WordNet *cousin* relation is encoded only for some words). Although the distinction between related vs. unrelated meanings is sometimes unclear, systematicity of the related senses among words is quite intuitive and has been well studied in Lexical Semantics (for example, (Apresjan, 1973; Nunberg, 1995; Copestake and Briscoe, 1995)). A comparison with WordNet cousin is discussed in the next section 4.

[9] Actually, cousin is one of the three relations which indicate the grouping of related senses of a word. Others are *sister* and *twin*. In this paper, we use cousin to refer to all relations listed in "cousin.tps" file (available in a WordNet distribution).

ually identified by the WordNet lexicographers.

To compare the automatically derived cluster pairs to WordNet cousins, we used the hypernym-hyponym relation in the trees, instead of the number or ratio of the overlapping words. This is because the levels at which the cousin relations are defined differ quite widely, from depth 0 to depth 6, thus the number of polysemous words covered in each cousin relation significantly varies. Therefore, it was difficult to decide on an appropriate threshold value for either criteria.

Using the hypernym-hyponym relation, we checked, for each cousin relation, whether there was at least one cluster pair that subsumed or was subsumed by the cousin. More specifically, for a cousin relation defined between nodes $c1$ and $c2$ in trees $T1$ and $T2$ respectively and a cluster pair defined between nodes $r1$ and $r2$ in the same trees, we decided on the correspondence if $c1$ is a hypernym or hyponym of $r1$, and $c2$ is a hypernym or hyponym $r2$ at the same time.

Based on this criteria, we obtained a result indicating that 173 out of the 194 cousin relations had corresponding cluster pairs. This makes the recall ratio 89%, which we consider to be quite high.

In addition to the WordNet cousins, our automatic extraction method discovered several interesting relations. Table 2 shows some examples.

5 A Lexicon based on Systematic Relations

Using the extracted cluster pairs, we partitioned word senses for all nouns and verbs in WordNet, and produced a lexicon. Recall from the previous section that our cluster pairs are generated for all possible binary combinations of basic classes, thus one sense could appear in more than one cluster pair. For example, Table 3 shows the cluster pairs (and a set of senses covered by each pair, which we call a *sense cover*) extracted for the noun "table" (which has 6 senses in WordNet). Also as we have mentioned earlier in section accuracy-result, our cluster pairs contain many false positives ones. For those reasons, we took a conservative approach, by disallowing transitivity of cluster pairs.

Table 3: Extracted Relations for "table"

Sense Cover	Cluster Pair	CN
(1 4)	[ARRANGEMENT, NAT_OBJ]	1.143
(1 5)	[ARRANGEMENT, SOC_GROUP]	1.143
(2 3)	[FURNITURE]	4.429
(2 3 4)	[FURNITURE, NAT_OBJ]	7.429
(2 3 5)	[FURNITURE, SOC_GROUP]	7.714
(2 3 6)	[FURNITURE, FOOD]	7.429
(4 5)	[NAT_OBJ, SOC_GROUP]	1.429
(5 6)	[SOC_GROUP, FOOD]	1.286

To partition senses of a word, we first assign each sense cover a value which we call a *connectedness*. It is defined as follows. For a given word w which has n senses, let S be the set of all sense covers generated for w. Let c_{ij} denote the number of sense covers in which sense i (s_i) and sense j (s_j) occurred together in S (where $c_{ii} = 0$ for all $1 \leq i \leq n$), and $d_{ij} = \sum_{k=1}^{n} \frac{c_{ik}+c_{kj}}{C}$, where $k \neq i$, $k \neq j$, $c_{ik} > 0$, $c_{kj} > 0$, and $C = \sum_{i,j} c_{ij}$. A connectedness of a sense cover $sc \in S$, denoted CN_{sc}, where $sc = (s_l, .., s_m)$ ($1 \leq l < m \leq n$) is defined as:

$$CN_{sc} = \sum_{i=l}^{m} \sum_{j=1}^{m} c_{ij} + d_{ij} \qquad (8)$$

Intuitively, c_{ij} represents the weight of a direct relation, and d_{ij} represents the weight of an indirect relation between any two senses i and j. The idea behind this connectedness measure is to favor sense covers that have strong intra-relations. This measure also effectively takes into account a one-level transitivity in d_{ij}. As an example, the connectedness of (2 3 4) is the summation of $c_{23}, c_{34}, c_{24}, d_{23}, d_{34}$ and d_{24}. Here, $c_{23} = 4$ because sense 2 and 3 cooccur in four sense covers, and $c_{34} = c_{24} = 1$. Also, $d_{23} = \frac{(c_{24}+c_{43})+(c_{25}+c_{53})+(c_{26}+c_{63})}{C} = \frac{2+2+2}{14} = .429$ (omitting cases where either or both c_{ik} and c_{kj} are zero), and similarly $d_{34} = .5$ and $d_{24} = .5$. Thus, $CN_{(234)} = 4 + 1 + 1 + .429 + .5 + .5 = 7.429$. Table 3 shows the connectedness values for all sense covers for "table".

Then, we partition the senses by selecting a set of non-overlapping sense covers which maximizes the total connectedness value. So in the example above, the set {(1 4),(2 3 5)} yields the maximum connectedness. Finally, senses that are not covered by any sense covers are taken as singletons, and added to the final sense partition. So the sense partition for "table" becomes {(1 4),(2 3 5),(6)}.

Table 4 shows the comparison between WordNet and our new lexicon. As you can see, our lexicon contains much less ambiguity: the ratio of monosemous words increased from 84% (88,650/105,461≈.84) to 92% (96,964/105,461≈.92), and the average number of senses for polysemous words decreased from 2.73 to 2.52 for nouns, and from 3.57 to 2.82 for verbs.

As a note, our lexicon is similar to CORELEX (Buitelaar, 1998) (or CORELEX-II presented in (Buitelaar, 2000)), in that both lexicons share the same motivation. However, our lexicon differs from CORELEX in that CORELEX looks at all senses of a word and groups *words* that have the same sense distribution pattern, whereas our lexicon groups

Table 2: Examples of Automatically Extracted Systematic Polysemy

Underspecified Class	Cluster Pair	Common Words
ACTION-LOCATION	[ACTION, POINT]	"drop", "circle", "intersection", "dig", "crossing", "bull's eye"
ARTIFACT-GROUP	[STRUCTURE, PEOPLE]	"house", "convent", "market", "center"
ARTIFACT-SUBSTANCE	[FABRIC, CHEMICAL_COMPOUND]	"acetate", "nylon", "acrylic", "polyester"
COMMUNICATION-PERSON	[VOICE, SINGER]	"soprano", "alto", "tenor", "baritone"
	[WRITING, RELIGIOUS_PERSON]	"John", "Matthew", "Jonah", "Joshua", "Jeremiah"

Table 4: WordNet vs. the New Lexicon

Category		WordNet	New
Nouns	Monosemous	82,892	88,977
	Polysemous	12,243	6,158
	Total words	95,135	95,135
	Ave # senses	2.73	2.52
Verbs	Monosemous	5,758	7,987
	Polysemous	4,568	2,339
	Total words	10,326	10,326
	Ave # senses	3.57	2.82
Total	Monosemous	88,650	96,964
	Polysemous	16,811	8,497
	Total words	105,461	105,461

word *senses* that have the same systematic relation. Thus, our lexicon represents systematic polysemy at a finer level than CORELEX, by pinpointing related senses within each word.

6 Evaluation: Inter-annotator Disagreement

To test if the sense partitions in our lexicon constitute an appropriate (or useful) level of granularity, we applied it to the inter-annotator disagreement observed in two semantically annotated corpora: WordNet Semcor (Landes *et al.*, 1998) and DSO (Ng and Lee, 1996). The agreement between those corpora is previously studied in (Ng *et al.*, 1999). In our current work, we first re-produced their agreement data, then used our sense partitions to see whether or not they yield a better agreement.

In this experiment, we extracted 28,772 sentences/instances for 191 words (consisting of 121 nouns and 70 verbs) tagged in the intersection of the two corpora. This constitutes the base data set. Table 5 shows the breakdown of the number of instances where tags agreed and disagreed.[10] As you can see, the agreement is not very high: only around 48%.[11]

Table 5: Agreement between Semcor and DSO

Category	Agree	Disagree	Total	Ave. κ
Nouns	6,528	5,815	12,343	.268
Verbs	7,408	9,021	16,429	.260
Total (%)	13,936 (48.4)	14,836 (51.6)	28,772 (100.0)	.264

This low agreement ratio is also reflected in a measure called the κ statistic (Carletta, 1996; Bruce and Wiebe, 1998; Ng *et al.*, 1999). κ measure takes into account chance agreement, thus better representing the state of disagreement. A κ value is calculated for each word, on a *confusion matrix* where rows represent the senses assigned by judge 1 (DSO) and columns represent the senses assigned by judge 2 (Semcor). Table 6 shows an example matrix for the noun "table".

A κ value for a word is calculated as follows. We use the notation and formula used in (Bruce and Wiebe, 1998). Let n_{ij} denote the number of instances where the judge 1 assigned sense i and the judge 2 assigned sense j to the same instance, and n_{i+} and n_{+i} denote the marginal totals of rows and columns respectively. The formula is:

$$k = \frac{\sum_i P_{ii} - \sum_i P_{i+}P_{+i}}{1 - \sum_i P_{i+}P_{+i}} \tag{9}$$

where $P_{ii} = \frac{n_{ii}}{n_{++}}$ (i.e., proportion of n_{ii}, the number of instances where both judges agreed on sense i, to the total instances), $P_{i+} = \frac{n_{i+}}{n_{++}}$ and $P_{+i} = \frac{n_{+i}}{n_{++}}$. The κ value is 1.0 when the agreement is perfect (i.e., values in the off-diagonal cells are all 0, that is, $\sum_i P_{ii} = 1$), or 0 when the agreement is purely

[10] Note that the numbers reported in (Ng *et al.*, 1999) are slightly more than the ones reported in this paper. For instance, the number of sentences in the intersected corpus reported in (Ng *et al.*, 1999) is 30,315. We speculate the discrepancies are due to the different sentence alignment methods used in the experiments.

[11] (Ng *et al.*, 1999) reports a higher agreement of 57%. We speculate the discrepancy might be from the version of WordNet senses used in DSO, which was slightly different from the standard delivery version (as noted in (Ng *et al.*, 1999)).

Table 6: Confusion Matrix for the noun "table" ($\kappa = .611$)

		Judge 2 (Semcor)							
		1	2	3	4	5	6	Total	
	1	43	0	0	0	0	0	43	$(= n_{1+})$
	2	6	17	3	0	0	0	26	$(= n_{2+})$
Judge 1	3	0	0	0	0	0	0	0	$(= n_{3+})$
(DSO)	4	1	0	0	0	0	0	1	$(= n_{4+})$
	5	0	0	0	0	0	0	0	$(= n_{5+})$
	6	2	2	1	0	0	0	5	$(= n_{6+})$
	Total	52	19	4	0	0	0	75	
		$(= n_{+1})$	$(= n_{+2})$	$(= n_{+3})$	$(= n_{+4})$	$(= n_{+5})$	$(= n_{+6})$	$(= n_{++})$	

Table 7: Reduced Matrix for "table" ($\kappa = .699$)

	1,4	2,3,5	6	Total
1,4	44	0	0	44
2,3,5	6	20	0	26
6	2	3	0	5
Total	52	23	0	75

Table 8: Our Lexicon vs. Random Partitions

Category	Total	Our Lexicon	Random
		Ave. κ	Ave. κ
Nouns	10,980	.247	.217
Verbs	14,392	.283	.262
Total	25,372	.260	.233

by chance (i.e., values in a row (or column) are uniformly distributed across rows (or columns), that is, $P_{ii} = P_{i+}P_{+i}$ for all $1 \leq i \leq M$, where M is the number of rows/columns). κ also takes a negative value when there is a systematic disagreement between the two judges (e.g., some values in the diagonal cells are 0, that is, $P_{ii} = 0$ for some i). Normally, $\kappa \geq .8$ is considered a good agreement (Carletta, 1996).

By using the formula above, the average κ for the 191 words was .264, as shown in Table 5.[12] This means the agreement between Semcor and DSO is quite low.

We selected the same 191 words from our lexicon, and used their sense partitions to reduce the size of the confusion matrices. For each word, we computed the κ for the reduced matrix, and compared it with the κ for a random sense grouping of the same partition pattern.[13] For example, the partition pattern of {(1 4),(2 3 5),(6)} for "table" mentioned earlier (where Table 7 shows its reduced matrix) is a multinomial combination $\binom{6}{2\ 3\ 1}$. The κ value for a random grouping is obtained by generating 5,000 random partitions which have the same pattern as the corresponding sense partition in our lexicon, then taking the mean of their κ's. Then we measured the possible increase in κ by our lexicon by taking the difference between the paired κ values for all words (i.e., κ_w by our sense partition - κ_w by random partition, for a word w), and performed a significance test, with a null hypothesis that there was no significant increase. The result showed that the P-values were 4.17 and 2.65 for nouns and verbs respectively, which were both statistically significant. Therefore, the null hypothesis was rejected, and we concluded that there was a significant increase in κ by using our lexicon.

As a note, the average κ's for the 191 words from our lexicon and their corresponding random partitions were .260 and .233 respectively. Those values are in fact lower than that for the original WordNet lexicon. There are two major reasons for this. First, in general, combining any arbitrary senses does not always increase κ. In the given formula 9, κ actually decreases when the increase in $\sum_i P_{ii}$ (i.e., the diagonal sum) in the reduced matrix is less than the increase in $\sum_i P_{i+}P_{+i}$ (i.e., the marginal product sum) by some factor.[14] This situation typically happens when senses combined are well distinguished in the original matrix, in the sense that, for senses i and j, n_{ij} and n_{ji} are 0 or very small (relative to the total frequency). Second, some systematic relations are in fact easily distinguishable. Senses in such relations often denote different objects in a context, for instance `ANIMAL` and `MEAT` senses of "chicken". Since our lexicon groups those senses together, the κ's for the reduce matrices decrease for the reason we mentioned above. Table 8 shows the breakdown of the average κ for our lexicon and random groupings.

[12] (Ng *et al.* 1999)'s result is slightly higher: $\kappa = .317$.

[13] For this comparison, we excluded 23 words whose sense partitions consisted of only 1 sense cover. This is reflected in the total number of instances in Table 8.

[14] This is because $\sum_i P_{i+}P_{+i}$ is subtracted in both the numerator and the denominator in the κ formula. Note that both $\sum_i P_{ii}$ and $\sum_i P_{i+}P_{+i}$ always increase when any arbitrary senses are combined. The factor mentioned here is $\frac{1-\sum_i P_{ii}}{1-\sum_i P_{i+}P_{+i}}$.

7 Conclusions and Future Work

As we reported in previous sections, our tree-cut extraction method discovered 89% of the WordNet cousins. Although the precision was relatively low (50-60%), this is an encouraging result. As for the lexicon, our sense partitions consistently yielded better κ values than arbitrary sense groupings. We consider these results to be quite promising. Our data is available at `www.depaul.edu/~ntomuro/research/naacl-01.html`.

It is significant to note that cluster pairs and sense partitions derived in this work are domain independent. Such information is useful in broad-domain applications, or as a *background lexicon* (Kilgarriff, 1997) in domain specific applications or text categorization and IR tasks. For those tasks, we anticipate that our extraction methods may be useful in deriving characteristics of the domains or given corpus, as well as customizing the lexical resource. This is our next future research.

For other future work, we plan to investigate an automatic way of detecting and filtering unrelated relations. We are also planning to compare our sense partitions with the systematic disagreement obtained by (Wiebe, *et al.*, 1998)'s automatic classifier.

Acknowledgments

The author wishes to thank Steve Lytinen at DePaul University and the anonymous reviewers for very useful comments and suggestions.

References

Apresjan, J. (1973). Regular Polysemy. *Linguistics*, (142).

Bruce, R. and Wiebe, J. (1998). Word-sense Distinguishability and Inter-coder Agreement. In *Proceedings of the COLING/ACL-98*, Montreal, Canada.

Buitelaar, P. (1998). CORELEX: Systematic Polysemy and Underspecification. Ph.D. dissertation, Department of Computer Science, Brandeis University.

Buitelaar, P. (2000). Reducing Lexical Semantic Complexity with Systematic Polysemous Classes and Underspecification. In *Proceedings of the ANLP/NAACL-00 Workshop on Syntactic and Semantic Complexity in Natural Language Processing*, Seattle, WA.

Carletta, J. (1996). Assessing Agreement on Classification Tasks: The Kappa Statistic, *Computational Linguistics*, **22**(2).

Copestake, A. and Briscoe, T. (1995). Semi-productive Polysemy and Sense Extension. *Journal of Semantics*, **12**.

Gonzalo, J., Chugur, I. and Verdejo, F. (2000). Sense Clusters for Information Retrieval: Evidence from Semcor and the InterLingual Index. In *Proceedings of the ACL-2000 Workshop on Word Senses and Multilinguality*, Hong-Kong.

Kilgarriff, A. (1997). Foreground and Background Lexicons and Word Sense Disambiguation for Information Extraction. In *Proceedings of the International Workshop on Lexically Driven Information Extraction.*

Kilgarriff, A. (1998a). SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs. In *Proceedings of the LREC.*

Kilgarriff, A. (1998b). Inter-tagger Agreement. In *Advanced Papers of the SENSEVAL Workshop*, Sussex, UK.

Landes, S., Leacock, C. and Tengi, R. (1998). Building Semantic Concordance. In *WordNet: An Electronic Lexical Database*, The MIT Press.

Li, H. and Abe, N. (1998). Generalizing Case Frames Using a Thesaurus and the MDL Principle, *Computational Linguistics*, **24**(2).

Miller, G. (eds.) (1990). WORDNET: An Online Lexical Database. *International Journal of Lexicography*, **3**(4).

Ng, H.T., and Lee, H.B. (1996). Integrating Multiple Knowledge Sources to Disambiguate Word Sense. In *Proceedings of the ACL-96*, Santa Cruz, CA.

Ng, H.T., Lim, C. and Foo, S. (1999). A Case Study on Inter-Annotator Agreement for Word Sense Disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Standardizing Lexical Resources*, College Park, MD.

Nunberg, G. (1995). Transfers of Meaning. *Journal of Semantics*, **12**.

Palmer, M. (1998). Are Wordnet sense distinctions appropriate for computational lexicons? In *Advanced Papers of the SENSEVAL Workshop*, Sussex, UK.

Pustejovsky, J. (1995). *The Generative Lexicon*, The MIT Press.

Rissanen, J. (1978). Modeling by Shortest Data Description. *Automatic*, **14**.

Tomuro, N. (2000). Automatic Extraction of Systematic Polysemy Using Tree-cut. In *Proceedings of the ANLP/NAACL-00 Workshop on Syntactic and Semantic Complexity in Natural Language Processing*, Seattle, WA.

Veronis, J. (1998). A Study of Polysemy Judgements and Inter-annotator Agreement. In *Advanced Papers of the SENSEVAL Workshop*, Sussex, UK.

Vossen, P., Peters, W. and Gonzalo, J. (1999). Towards a Universal Index of Meaning. In *Proceedings of the ACL SIGLEX Workshop on Standardizing Lexical Resources*, College Park, MD.

Wiebe, J., Bruce, R. and O'Hara, T. (1999). Development and Use of a Gold-Standard Data Set for Subjectivity Classifications. In *Proceedings of the ACL-99*, College Park, MD.