Regular Polysemy in WordNet and Pattern based Approach

Abed Alhakim Freihat, Fausto Giunchiglia
Dept. of Information Engineering and Computer Science
University of Trento,
Trento, Italy
e-mail: {fraihat,fausto}@disi.unitn.it

Biswanath Dutta
Documentation Research and Training Centre
Indian Statistical Institute (ISI)
Bangalore, India
e-mail: bisu@drtc.isibang.ac.in

***Abstract*— WordNet represents the polysemous terms by capturing the different meanings of them at lexical level implicitly without giving emphasis on the polysemy types they belong to. This problem affects the usability of WordNet as a suitable knowledge representation resource for Natural Language Processing applications. The current work presents pattern based approach for solving the polysemy problem by transforming the implicit relations between the synsets at lexical level into explicit relations at the semantic level.**

***Keywords*— lexical databases; WordNet; specialization polysemy; metaphoric polysemy; homonymy; regular polysemy; polysemy reduction; lexical semantics; semantic search; knowledge engineering**

I. INTRODUCTION

Solving the polysemy problem is very crucial in many research fields including machine translation, information retrieval and semantic search [1] since polysemy in WordNet [2] is considered to be the main reason that makes it hard to use for natural language processing (NLP) and semantic applications.

Polysemy corresponds to various kinds of linguistic phenomena and belongs to different polysemy classes. Recognizing the polysemy class of a given polysemous term is essential in NLP since different polysemy phenomena require different processing strategies. Differentiating between the polysemy classes should be possible through explicit semantic relations between the senses of polysemous terms. Unfortunately, relations between polysemous terms are not provided in WordNet [3]. For instance, WordNet does not provide the distinction between homographs, and complementary terms [4].

In the last decades, many approaches have been introduced to solve the polysemy problem through merging the similar meanings of polysemous terms [5]. These approaches are sometimes helpful in cases, where terms have meanings that are similar enough to be merged. However, polysemous terms with similar meanings are a sub-case of the solution of specialization polysemy [6]. They represent only a small portion of the polysemy problem. In fact, a significant portion of the polysemous senses should not be merged, as they are just similar in meaning [7] and not redundant. In another approach, CORELEX [4] has been introduced as an ontology of systematic polysemous nouns extracted from WordNet. However, CORELEX deals only with the upper level ontology of WordNet that corresponds mainly to the metonymy cases and does not provide a solution for other polysemy types [6].

In this paper, we introduce a pattern based approach that combines several ideas to solve the polysemy problem. Our approach follows the idea that the polysemy problem is a problem of semantic organization [9]. Thus, the goal of our approach is to reorganize the semantic structure of the polysemous terms in wordNet, where we transform the implicit relations between the polysemous terms at lexical level to explicit relations at the semantic level. This includes extending WordNet by adding new hierarchical and associative relations between the synsets to explicitly denote the polysemy type occurring between the meanings of each polysemous term, as suggested in [3]. To achieve this goal, our approach deals with all polysemy types at all ontological levels of WordNet. It deals with the lower level ontology of WordNet and it extends the merge operation suggested by the polysemy reduction approaches [5][10] by providing new operations that organize the relations between the meanings of polysemous terms. Our approach also deals with polysemy in the middle level, as it is the case in regular polysemy approaches [11] and also in the upper level ontology as in systematic polysemy approaches [4].

This paper is organized as follows. In Section II, we discuss the polysemy problem in wordNet. In Section III, we describe the current approaches for solving the polysemy problem in WordNet. In Section IV, we present the semantic relations that denote polysemy types and the operations that reorganize the structure of polysemous terms in WordNet. In Section V, we introduce a pattern based approach for solving the polysemy problem in the case of polysemous nouns. In Section VI, we discuss the rules that we use by reorganizing the ontological structure of polysemous terms. In Section VII, we discuss the results and evaluation of our approach. In Section VIII, we conclude the paper and describe our future research work.

II. POLYSEMY IN WORDNET

WordNet is a lexical database that organizes synonyms of English words into sets called synsets, where each synset is described through a gloss. For example, the words *happiness* and *felicity* are considered to be synonyms and are grouped into a synset {*happiness, felicity*} that is described through the gloss*: state of well-being* *characterized by emotions ranging from contentment to intense joy*.

WordNet organizes the relations between synsets through semantic relations, where each word category has a number of relations that are used to organize the relations between the synsets of that grammatical category. For example, the hyponymy relation (X is a type of Y) is used to organize the ontological structure of nouns. WordNet 2.1 contains 147,257 words, 117,597 synsets and 207,019 word-sense pairs. Among these words there are 27,006 polysemous words, where 15,776 of them are nouns.

From linguistics, a term is polysemous if it has more than one meaning[19]. Linguists differentiate between contrastive polysemy, i.e., terms with completely different and unrelated meanings - also called homonyms or homographs; and complementary polysemy, i.e., terms with different but related meanings. Complementary polysemy is classified in three sub types: metonymy, specialization polysemy and metaphors. Following the above, we can classify the various forms of polysemy as follows:

1) Complementary polysemy: terms that have the same spelling and related meanings. Complementary polysemy can be:
   a. Metonymy: substituting the name of an attribute or feature for the name of the thing itself, such as in the following example the term *chicken*:
      Peter caught *a chicken* in his garden.
      Peter prepared *chicken* for the dinner.
   b. Specialization polysemy: a term is used to refer to a more general meaning and another more specific meaning, such as in the following example the term *methodology*:
      `#1` **`methodology,`** `methodological analysis: the branch of philosophy.`
      `#2` **`methodology`**`: the system of methods followed in a particular discipline.`
   c. Metaphors: terms that have the same spelling and have literal and figurative meanings. Consider, for instance, the term *parasite*:
      `#1` **`parasite`**`: an animal or plant that lives in or on a host (another animal or plant).`
      `#2 leech,` **`parasite`**`, sponge, sponger: a follower who hangs around a host (without benefit to the host) in hope of gain or advantage.`
2) Homographs: terms that have the same spelling and different unrelated meanings, such as in the following example the term *bank*:
   Peter sat on the *bank* of the river.
   Peter deposited money in the *bank*.

In WordNet, the number of senses for a polysemous term may range from 2 to more than 30. In some rare cases, the number of senses is even more. For instance, the noun *head* has 33 senses. Nevertheless, 90% of the polysemous terms are nouns. Table I shows the distribution of these polysemous nouns according to the number of senses they have. Notice that, in this paper, we are concerned with polysemous nouns only, and not the verbs, adverbs and adjectives.

The fact that a term has more than two senses implies that the meanings of the term belong to more than one type of polysemy. For example, the term *food* has 3 senses as mentioned below, where the polysemy type between the first and the second meanings is specialization polysemy, while the third meaning is metaphoric.

TABLE I. POLYSEMOUS NOUNS IN WORDNET

# of senses	# of nouns (in percentage)
2	9328 (≈ 64.2%)
3	2762 (≈ 19%)
4	1083 (≈ 7.4%)
5	555 (≈ 3.8%)
6	277 (≈ 1.9%)
7	194 (≈ 1.3%)
8	90 (≈ 0.7%)
9	88 (≈ 0.6%)
10	54 (≈ 0.37%)
>10	94 (≈ 0.64%)
Total	14525 (=100%)

`#1` **`food`**`, nutrient: any substance that can be metabolized by an organism to give energy and build tissue.`
`#2` **`food`**`, solid food: any solid substance that is used as a source of nourishment.`
`#3` **`food`**`, food for thought: anything that provides mental stimulus for thinking.`

III. APPROACHES FOR SOLVING POLYSEMY IN WORDNET

The approaches of polysemy can be classified in three main approaches. The first is polysemy reduction, where the focus is on complementary polysemy to produce more coarse-grained lexical resources of existing fine-grained ones, such as WordNet. The second type of polysemy approaches focuses on classifying polysemy into systematic or regular polysemy and homographs. Based on this classification, CORELEX was introduced as ontology of systematic polysemous nouns extracted from WordNet. The third type of polysemy approaches is semantic relations extraction approaches. These approaches propose to enrich wordNet with semantic relations that correspond to the implicit relations between the complementary polysemous terms in WordNet.

In the following, we summarize polysemy reduction approaches, CORELEX, and the most prominent semantic relations extraction approaches. Notice that neither polysemy reduction approaches nor systematic polysemy approaches could solve the polysemy problem in WordNet. In general, polysemy reduction approaches could not solve the problem of the upper level ontology, while systematic polysemy approaches did not provide a solution for the polysemy problem in the middle and lower level ontology of WordNet.

A. *Polysemy Reduction Approaches*

In polysemy reduction, the senses are clustered such that each group contains related polysemous words [10][14]. These groups are called homograph clusters. Once the clusters have been identified, the senses in each cluster are merged. To achieve this task, several strategies have been introduced. These strategies can be mainly categorized in semantic-based and statistics-based strategies [15]. Some approaches combine both strategies [10]. Although results of applications of these approaches are reported, these results are taken usually from applying them on sample data sets, and there is no way to verify these results independently. Polysemy reduction approaches typically rely on the application of some detection rules such as: *if S1 and S2 are two synsets containing at least two words, and if S1 and S2 contain the same words, then S1 and S2 can be collapsed together into one single synset* [10]. However, applying this rule may wrongly result in merging two different senses as in the following example:

#1 **smoke, smoking***:* a hot vapor containing fine particles of carbon being produced by combustion.
#2 **smoke, smoking***:* the act of smoking tobacco or other substances.

In general, polysemy reduction can neither predict the polysemy type occurring between the senses of polysemous words nor can deal with metonymy or metaphors. Polysemy reduction does not solve the polysemy problem in linguistic resource. Nevertheless, it can be potentially used to solve part of the problem, namely, the identification and merging of genuine redundant synsets.

B. *CORELEX*

J. Apresjan defined regular polysemy as follows: "*a polysemous Term T is considered to be regular if there exists at least another polysemous T' that is semantically distinguished in the same way as T*" [16]. CORELEX and regular polysemy approaches in general rely on this definition. These approaches follow two different methods to solve the polysemy problem in WordNet:

CORELEX, the first systematic polysemy lexical database, follows the generative lexicon theory [9] that distinguishes between systematic (also known as regular or logic) polysemy and homographs. Systematic polysemous words are systematic and predictable while homonyms are not regular and not predictable. The type of polysemy of the word *fish*, for example, is systematic since the meaning *food* can be predicted from the *animal* meaning, and so the word *fish* belongs to the systematic class *animal food*. The two meanings of fish describe two related aspects of *fish*: fish is an animal and fish is a food. A word is systematic polysemous means that the meanings of this word are not homonyms and they describe different aspects of the same term. Following this distinction, CORELEX organizes the polysemous nouns of WordNet 1.5 into 126 systematic polysemy classes. The systematic polysemy classes in CORELEX have been determined in a top-down fashion considering the patterns in the upper level ontology of wordNet only [11]. The high level basic types in CORELEX patterns make them too coarse grained to extract useful semantic relations [4][11][18]. At the same time, there are hundreds of regular structural patterns that reside in the middle level and lower level ontology that are not covered by the high level basic types. These patterns correspond to metaphoric and specialization polysemy [1][4]. The underspecification method is not appropriate to CORELEX patterns that correspond to metaphoric polysemy. CORELEX patterns contain too many false positives [11][18]. Another important point is related to the fine grained nature of WordNet, where the meanings of some CORELEX classes are very difficult to disambiguate, and indistinguishable even for humans [13].

C. *Semantic Relations Extraction Approaches*

The semantic relations extraction approaches are regular polysemy approaches that attempt to extract implicit semantic relations between the polysemous senses via regular structural patterns. The basic idea in these approaches is that the implicit relatedness between the polysemous terms corresponds to variety of semantic relations. Extracting these relations and making them explicitly should improve wordNet [11][18]. These approaches refine and extend CORELEX patterns to extract the semantic relations. Beside the structural regularity, these approaches exploit also the synset gloss [4][11] and the cousin relationship [11][18] in WordNet. For example, the approach described in [4] exploits synset glosses to extract *auto-referent candidates*. The approach described in [18] uses several strategies, such as *ontological bridging* to detect relations between the sense pairs. In general, the extracted relations in these approaches are similar. For example, we find the relations *similar to*, *color of* in the results of the approach in [4]. The result in [11] contains relations such as *contained in*, *obtain from*. Similarly, the result in [18] contains relations such as *fruit of*, *tree of*.

The semantic relations extraction approaches are in general better than CORELEX in the following aspects. First of all, the discovered patterns in these approaches are more fine grained and enable to capture meaningful relations. These approaches classified the complementary polysemy in three sub classes: metonymy, metaphoric, and specialization polysemy. Another important point in these approaches is that these approaches considered the problem of false positives. Anyway, these approaches did cover only few patterns of the specialization polysemy and metaphoric cases. They did not address the problem of too fine grained senses or discourse dependent polysemy.

In this section, we have discussed some of the state of the art approaches for solving the polysemy problem in

WordNet. Our primary observation here is that these approaches complement each other. The same holds for the approach presented in this paper. Our approach is not an alternative for the presented solutions. Instead, we see our approach as a complementary solution for the state of the art solutions that focuses on the specialization polysemy problem. Solving the specialization polysemy problem is important complementary step for the presented solutions, because solving the specialization polysemy problem enhances the usability of WordNet as a knowledge representation resource. Solving the specialization polysemy problem includes solving the following problems in WordNet:

- **The problem of implicit relatedness:** The implicit relatedness between specialization polysemy cases is a hierarchical. For example, representing the hierarchical relatedness between the senses of the term `white croaker` from knowledge representation point of view is more appropriate than representing at the lexical level only.
  `#1` **`white croaker`**`, queenfish, Seriphus politus: silvery and bluish fish of California.`
  `#2` **`white croaker`**`, kingfish, Genyonemus lineatus: silvery fish of California.`
- **The problem of too fine grained senses:** WordNet contains a reasonable amount of too fine grained senses. For example, the first sense of the term `optimism` corresponds to `optimistic feeling` and the second meaning corresponds to `disposition`.
  `#1 optimism: the optimistic feeling that all is going to turn out well.`
  `#2 optimism: a general disposition to expect the best in all things.`

  Capturing the difference between the meanings of such cases is very difficult. In some cases, the too fine grained distinction between senses may result in redundancy as in the following example.
  `#1 lullaby, cradlesong, berceuse: a quiet song intended to lull a child to sleep.`
  `#2 lullaby, cradlesong: a quiet song that lulls a child to sleep.`
- **The problem of discourse dependent polysemy:** WordNet contains a reasonable amount of discourse dependent polysemy cases. For example, using the term `center` to refer to the following meanings cannot be understood without a proper context.
  `#2 center field, centerfield, center: the piece of ground in the outfield directly ahead of the catcher.`
  `#6 center, center of attention: the object upon which interest and attention focuses.`
  `#7 center, centre, nerve center, nerve centre: a cluster of nerve cells governing a specific bodily process.`
  `#15 plaza, mall, center, shopping mall, shopping center, shopping centre: mercantile establishment consisting of a carefully landscaped complex of shops representing leading merchandisers.`

It is not clear, which rule wordNet is following by adding such discourse dependent terms in the synset synonyms. In this example, it is not clear, why wordNet considers the term `center` to be a synonym in the previous cases and it does not consider it a synonym of the terms `city center`, `medical center`, or `research center`. In addition to these problems, our approach is able to discover homonymy and metaphoric cases that reside in the middle level or lower level ontology of wordNet. We provide a solution for the discovered homonymy and metaphoric cases as explained in the next section.

IV. Denoting Polysemy Types and Organizing Polysemy in WordNet

A. Polysemy Type Relations

In the following, we explain the suggested relations to denote the polysemy types:

Homographs: There is no relation between the senses of a homograph term. Nevertheless, differentiating homographs from other polysemy types is very important improvement in wordNet. We use the relation *is_homograph* to denote that two synsets of a polysemous term are homographs. For example, this relation holds between the synsets *{saki as alcoholic drink}* and *{saki as a monkey}*.

Metonymy: In metonymy cases, there is always a *base meaning* of the term and other *derived meanings* that express different aspects of the base meaning [17]. For example, the term chicken has the base meaning {*a domestic fowl bred for flesh or eggs*} and a derived meaning {*the flesh of a chicken used for food*}. To denote the relation between the senses of a metonymy term, we use the relation *has_aspect*, where this relation holds between the base meaning of a term and the derived meanings of that term. To set up the relation we need to determine the base meaning and then relate the other derived meanings to it.

Metaphors: In metaphoric cases, we use the relation *Is_metaphor* to denote the metaphoric relation between the metaphoric meaning and literal meaning of a metaphoric term. For example, this relation is used to denote that {*cool as great coolness and composure under strain*} is metaphoric meaning of the literal meaning {*cool as the quality of being at a refreshingly low temperature*}. In the cases where this relation is applicable, we need to specify the literal meaning and the metaphoric meaning.

B. Operations for Specialization polysemy

Analysis of specialization polysemy cases shows that such cases can be classified based on the synset synonyms into the following three groups. To explain our idea, we have chosen cases, where the synsets of each term share the same common parent.

Let T be a polysemous term that occurs in two synsets S_1 and S_2. We consider T in the following three cases:

Case 1: T has synonyms in S_1 and has synonyms in S_2 as in the case of `kestrel`:

```
#1 sparrow hawk, American kestrel, kestrel, Falco sparverius: small American falcon.
#2. kestrel, Falco tinnunculus: small Old World falcon.
```

Case 2: T has synonyms in S_1 or in S_2 but not in both as in the case of *dorsum*:

```
#1 back, dorsum: the posterior part of a human (or animal) body from the neck to the end of the spine.
#2 dorsum: the back of the body of a vertebrate or any analogous surface.
```

Case 3: T has no synonyms in S_1 or S_2 as the in the case of *compatible software*:

```
#1 compatible software: application software programs that share common conventions.
#2 compatible software: software that can run on different computers without modification.
```

In case 1, T has synonyms in S_1. This means that T is exchangeable with the other synonyms of S_1 and at the same time is also exchangeable with the synonyms of S_2. Let T_1, T_2 be non polysemous synonyms of T in S_1 and S_2 respectively. T_1 is synonymous with T but not with T_2. Otherwise, T_1 and T_2 should appear in the same synset. The fact that T_1 and T_2 appear in two different sibling synsets indicates that they are not the same. We think that the semantic relatedness between S_1 and S_2 is encoded at lexical level rather than at semantic level. We have the same observation in case 2. The fact that one synset contains T only and the other synset contains additional terms indicates that the synset that contains T only is a more general meaning of the synset that have additional terms. We consider the terms in case 3 as candidates to be merged. Accordingly, we suggest the following operations to organize the relations between the senses in specialization polysemy cases.

Solution for Case 1: We add a new (missing) parent in cases where the polysemous meanings of a term T can be seen more specific meanings of an absent more general meaning. In such cases, we create the missing parent, which is a more general meaning and connect the more specific meanings to this newly created parent. This operation is schematized in Figure 1.

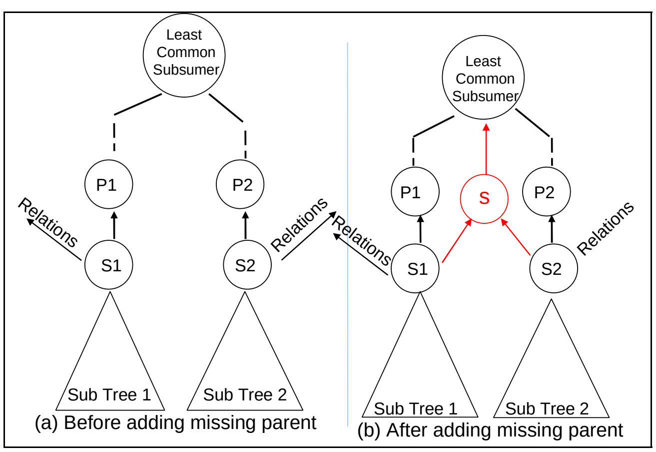

Figure 1. Adding missing parent

Solution for Case 2: We establish a new (missing) *is_a* relation to denote that a sense of a polysemous term T is more specific than another more general meaning of T. We schematize this operation as illustrated in Figure 2.

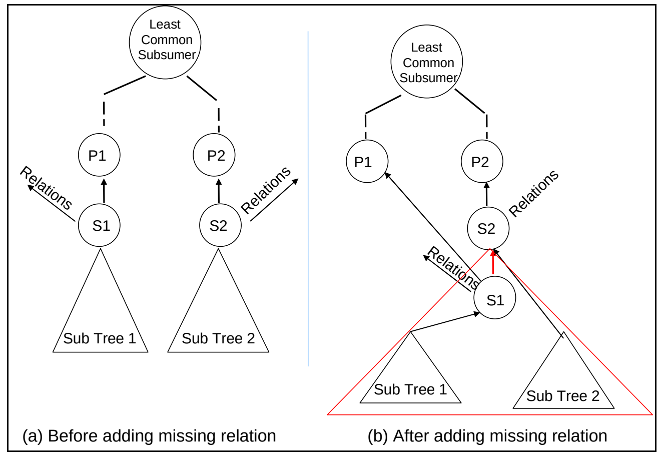

Figure 2. Adding missing relation

Solution for case 3: We merge the meanings. The merge operation is schematized as in Figure 3.

At the term level, we disambiguate the polysemous terms as follows: in case 1, we remove the polysemous terms from both child synsets and keep the polysemous words in the new added parent synset only. In case 2, we remove the polysemous term from the synset with the more specific meaning and keep it in the synset with the more generic meaning. The Merge operation in case 3 unifies the terms of both synsets in one synset. Thus, applying the three operations results in reducing the number of polysemous words in WordNet.

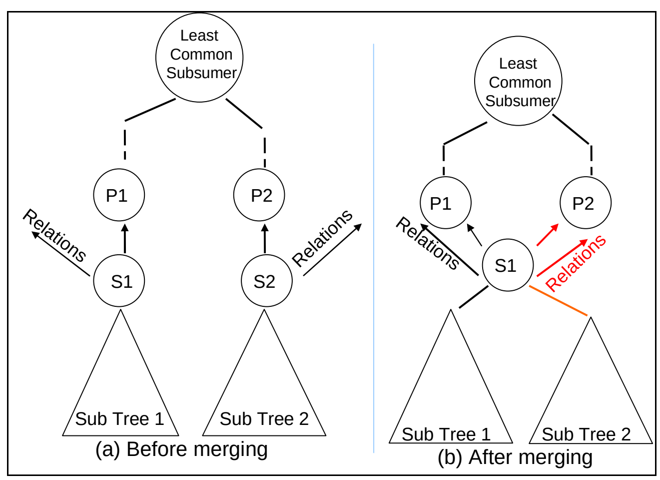

Figure 3. Merge operation

V. PATTERN BASED APPROACH FOR SOLVING POLYSEMY

In this section, we describe our approach for solving polysemy in WordNet. The approach has the following five phases. Phases A, C, and E are automatic, while B and D are manual.

A. Patterns Identification
B. Patterns Classification
C. Polysemy type Assignment
D. Validation
E. Applying the polysemy relations and operations

A. Patterns Identification

In this phase, we discuss the algorithm that is used to identify the regular type compatible patterns. Before describing the algorithm, we illustrate the definitions we used in the algorithm.

Definition 1 (Term). A term T is a triple <Label, Cat, Rank>, where

a) Label is the term label, i.e., a word which is the orthographic string representation of the term;
b) Cat is the grammatical category of the term;
c) Rank is the term rank, i.e., a natural number > 0.

Wordnet organizes terms into synsets, where each synset contains an ordered list of synonymous terms. We define wordNet synsets as follows.

Definition 2 (wordNet synset) A synset S is a quintuple <Terms, Label, Gloss, Rels, Rank>, where

a) Terms is an ordered list of synonomous terms that have the same grammatical category, called synset synonyms;
b) The grammatical category of the synset is the grammatical category of its terms;
c) The term rank of the synset terms corresoponds to its position in the terms list;
d) Label $\in$Ts is the synset label, i.e., the prefferd term of the synset is the first term in the terms list;
e) Gloss is a natural language text that describes the synset;
f) Rels is a set of semantic relations that hold between the synsets;
g) Rank is the synset rank, i.e., a natural number > 0 that reflects the familiarity of the synset.

The set Rels in the previous definition correspond to the semantic relations used by WordNet to organize the relations between the synsets. In the following, we define the relation hypernym and hyponym, the counter relation of hypernym.

Definition 3 (hypernym/hypernym relation). The relations hypernym/hyponym are hierarchical relations in WordNet that denote the superordinate/subordinate realtionship betwen synsets. For two synsets s_1, s_2: s_1 is a hypernym of s_2 is equivalent to s_2 is a hyponym of s_1.

Using the hypernym relation, wordNet organizes synsets in the case of nouns in a hierarchy. We define the hierarchy of WordNet in the case of nouns as follows:

Definition 4 (wordNet hierarchy). WordNet hierarchy WH is a rooted diagraph <N, E>, where

a) N is a set of synsets that belong to the grammatical category noun;
b) Entity $\in$ N is the root of WH;
c) $E \subseteq N^2$;
d) $(s_1, s_2) \in E$ if s_1 is a hypernym of s_2;
e) $\forall\, s((s \in N \wedge s \neq \text{entity}) \Rightarrow \exists s'((s', s) \in E))$.

A term is considered to be polysemous if it is found in the terms of more than one synset. We call such synsets polysemous synsets. It is possible for two polysemous synsets to share more than one term. Two polysemous synsets and their shared terms constitute a polysemy case. In the following, we define a polysemy case as follows.

Definition 5 (polysemy case) A polysemy case is a triple <Ts, s_1, ..., s_n>, where s_1, ..., s_n are polysemous synsets that have the terms Ts in common.

Note that the polysemy cases c_1 = <Ts, s_1, s_2> and c_2 = <Ts, s_2, s_1> are considered to be one polysemy case. We exploit the hypernym relation to discover the terms that are "*semantically distinguished in the same way"* as stated in Apresjan's definition. We consider polysemy cases to be semantically distinguished in the same way if they have the same structural pattern. In the following, we define structural patterns.

Definition 6 (Structural Pattern) Let c=<Ts, s_1, ..., s_n> be a polysemy case. Let *R* be a subset of {s_1, ..., s_n}. Let *Q* an ordered sequence of *R*, where $|R| = m, 2 \leq m \leq n$, and $Q = < s_1,..,s_m >, s_i \in R, s_i \neq s_j$, for $i \neq j$. A structural pattern is defined as $p\# < p_1,.., p_m >$, such that each p_i is a direct hyponym of *p* and subsumes $s_i, 1 \leq i \leq m$. We call *p* the pattern head and p_i the pattern parts.

Definition 7 (Regular Structural Pattern) A pattern is regular if there are at least two terms that belong to it.

For example, the pattern *passerine#<oscine, tyrannid>* is regular since there are 3 terms that belong to it.

Definition 8 (Sub pattern) For a regular pattern *ptrn* = $p\# < p_1,.., p_m >$. A pattern *ptrn'* is a sub pattern of *ptrn* if $ptrn' = p\# < p_1',..,p_k' >$ and $\exists p_i, p_j' (p_i = p_j')$.

Sub patterns are important, since it is possible that the elements of a pattern and its sub patterns have the same polysemy type. For example, the pattern *passerine#<oscine,*

tyrannid> and its sub pattern *passerine#<oscine,wren>* both belong to the specialization polysemy patterns.

Definition 9 (Common parent class) A term belongs to the common parent class if it has at least two synsets that share the same hypernym.

For example, the synsets of the term *kestrel* in the previous section share the same hypernym. In polysemy reduction approaches, senses that have the common parent property are candidates to be merged. In our approach, such terms are candidates for specialization polysemy. Note that there are many terms that have this property, but they are not considered to be regular according to definition 1, since they have different hierarchical structures.

In the following, we explain the pattern extraction algorithm.

Algorithm : Regular Polysemy Patterns Extraction
Input:
PNOUNS = Polysemous nouns in WordNet
UR = the list of the unique beginners in WordNet
SNR = the number of the term synsets,
Output:
N = an associative array to store the regular patterns.
M = an associative array to store the sub patterns
P = a list to store the elements of the common parent class
O = a list of singleton patterns

1. *poly_nouns* = retrieve_polysemous_nouns(*SNR*);
2. **For each** *noun* **in** *poly_nouns*
3. *S* = retrieve_synsets(*noun*);
4. *ptrns* = construct_patterns(S);
5. **For each** $Q \subseteq S$
6. **If** $Q \in$ *Common Parent*
7. add <*noun, Q*> to P;
8. **For each pattern** *ptrn* = $p\#\langle p_1,..,p_m\rangle$ **in** *ptrns*
9. **If** $p \notin UR$
10. Add *noun* to the list under *ptrn* in N;
11. **For each** *ptrn* **in** N
12. **If** |N[*ptrn*]| > 1
13. M[ptrn] = sub_patterns(*ptrn*)
14. Remove sub_patterns(*ptrn*) from N
15. **For each** *ptrn* **in** N
16. **If** |N[*ptrn*]| < 2
17. Add *ptrn* to O;
18. Remove *ptrn* from N;
19. **return** <N,M,P,O>;

The presented algorithm works in three phases:

1. **Patterns and common parent terms identification (line 1 to 10):** We retrieve the list of all nouns that have the sense number given in the algorithm input. We check, whether the term belongs to the common parent class and also whether it has regular patterns. We exclude the top level ontology patterns such as *physical entity#<physical object, physical process>*. Such patterns correspond usually to CORELEX patterns and they are not specialization polysemy patterns. Notice also that it is possible for terms that have more than 2 senses to have more than one pattern. The function **construct_patterns** is explained below.
2. **Sub patterns identification (lines 11 to 14):** If more than one term belongs to a pattern, it is a regular pattern, and then we search all singleton patterns to identify possible sub patterns of that pattern. Identified sub patterns are removed from the patterns list and added to the sub patterns list.
3. **Singleton patterns identification (lines 15 to 18):** After identifying the sub patterns, the remaining singleton patterns are removed from the patterns list and added to the list of the singleton patterns.

In the following, we explain the algorithm **construct_patterns** that is used for constructing patterns.

Algorithm: construct_patterns
Input:
Synsets : a list of synsets
Output: a list of patterns

1. parents := a list of synsets;
2. **For each** synset s1 **in** synsets
3. **For each** synset s2 ≠ s1 **in** synsets
4. parent = getLeastCommonSubsumer(s1, s2);
5. **If** parent ∉ parents
6. add parent to parents;
7. patterns : a list of pattern strings;
8. **For each** parent **in** parents
9. ptrnSynsets = an empty list of synsets;
10. **For each** synset syn **in** Synsets
11. **If** isHypernym(parent, syn)
12. add syn to ptrnSynsets;
13. **If** |ptrnSynsets| > 1
14. ptrnLabel=constructPtrnLabel(parent, ptrnSynsets);
15. add ptrnLabel to patterns;
16. **return** patterns;

The algorithm construct_patterns works as follows.

1. **Computing possible pattern heads (line 2 to 6):** We compute the possible pattern heads by computing the least common subsumer of the synsets.
2. **Grouping the pattern synsets (line 7 to 15):** We compute the synsets that belong to the patterns according to the pattern heads constructed in the previous step.
3. **Constructing the pattern label (line 14):** We use the function constructPtrnLabel that constructs the pattern label of a pattern.

The algorithm **constructPtrnLabel** is explained below.

Algorithm: constructPtrnLabel
Input:
ptrnSynsets: a list of synsets that belong to a pattern

parent: the pattern head of the synsets in ptrnSynsets
Output: patternLabel

```
1. ptrnLabel : a string;
2. ptrnLabel = parent."#<";
3. sort ptrnSynsets;
4. For i = 0; I < |ptrnSynsets| -1
5.      Synset s  = ptrnSynsets[i];
6       p = getRootHypynomRelativeToSynset(parent, s);
8.      ptrnLabel .= p. ",";
9.   Synset s  = ptrnSynsets[|ptrnSynsets|-1];
10. p = getRootHypynomRelativeToSynset(parent, s);
11. ptrnLabel .= p . ">";
12. return ptrnLabel;
```

The algorithm constructs the pattern label as defined in definition 1. The synsets are sorted alphabetically to ensure that the pattern label is a unique identifier of the pattern.

The results of applying the algorithm on the polysemous terms in WordNet are as follows: the total number of the nouns in WordNet is 14525 nouns. The algorithm identified 12565 polysemy cases to belong to type compatible patterns. The algorithm returned four lists: a pattern list that contains 1169 patterns, a sub patterns list that contains 2855 sub patterns, the list of the common parents that contains 1002 cases, and a list that contains 700 singleton patterns. The average time to generate the patterns on a single computer is about 45 minutes.

The algorithm returns the following lists:
1. **a list of regular patterns:** contains the regular patterns, where at least two terms belong to each pattern.
2. **a list of sub patterns:** contains the sub patterns of the patterns identified in the regular patterns list.
3. **a list of common parent terms:** contains the terms, where the synsets or part of the synsets of these terms share the same hypernym.
4. **a list of singleton patterns:** this list contains the patterns that have less than two terms and are not sub patterns of any regular pattern. Notice that it is possible for terms that have more than 2 senses to have more than one pattern.

B. *Patterns Classification*

Our task in this phase is to classify the patterns in specialization polysemy and metaphoric polysemy. First of all, the terms that belong to the common parent are considered as specialization polysemy candidates. We consider also the polysemy type of the sub patterns as the polysemy type of the pattern, they belong to. To classify the patterns, we have arranged them into hierarchies. Some examples for the roots of the hierarchies are shown in Table II. The numbers rights to the types correspond to the number of patterns that belong to that type.

Analyzing the patterns under these types shows that these patterns can be classified into four groups:

1. Specialization polysemy patterns
2. Metaphoric patterns
3. Homonymy patterns
4. Mixed patterns

TABLE II. EXAMPLES FOR TYPE COMPATIBLE PATTERNS ROOTS IN WORDNET

Patterns under physical entity		Patterns under abstract entity	
Type	#patterns	Type	#patterns
substance	19	psychological feature	1
organism	9	cognition	19
person	148	attribute	13
animal	5	communication	27
plant	6	measure	11
artifact	90	group	17
process	10	time period	6
location	7	relation	11
thing	5	act	70

In the following, we explain our criteria by classifying the patterns.

1. Specialization Polysemy patterns: the type of some specialization polysemy patterns can be determined directly by considering the type of the pattern only. For example, it is clear that the patterns whose type belongs to *animal* and the types under *animal* are specialization polysemy, or at least, it is not common at all to find a metaphoric link between the types under *animal*. The criterion for determining other specialization polysemy patterns is the *consistency* of the pattern subtypes.

2. Metaphoric patterns: to determine metaphoric patterns, we followed the idea that metaphors are human centric in the sense that we use metaphors to express our feelings, judgments, situations, irony, and so on. For example, when we use *sponger* to refer to some one, we are making a judgment upon that person. This gives us a hint, where to search for metaphoric patterns, namely, under the person type, or the types whose subtypes indicate meaning transfer from their literal meaning to a (metaphoric) human centric meaning as discussed below. Here, the type attribute is an example of such cases.

a. Metaphoric patterns under person: we found under the type person 106 patterns. Some of these patterns are specialization polysemy patterns and others are metaphoric. To determine metaphoric patterns under the type person, we searched for inconsistency between the subtypes of the patterns. For example, we find such inconsistency in the pattern person#<bad person, worker>. The subtype bad person is not consistent with the type worker and, therefore a specialization polysemy is totally excluded in this pattern. The term iceman is an example of terms that belong to this pattern:

```
#1 iceman:  someone who cuts and delivers ice.
#2 hatchet man, iceman: a professional killer.
```

On the other hand, the subtypes of the pattern person#<expert, worker> are consistent and is considered as a specialization polysemy pattern. The term technician is an example for this pattern:

```
#1 technician: someone whose occupation involves
training in a technical process.
#2 technician: someone known for high skill in
some intellectual or artistic technique.
```

b. Metaphoric patterns under attribute: our criteria here were to find meaning transfer between the subtypes. Attribute has the following four patterns: attribute#<property, trait>, attribute#<property, state>, attribute#<property, quality>, and attribute#<quality, trait>, with the following meanings:

```
Property: a basic or essential attribute shared by
all members of a class.
State: a state of depression or agitation.
Quality: an essential and distinguishing attribute
of something or someone.
Trait: a distinguishing feature of your personal
nature.
```

The meaning transfer from property to human centric meaning is clear in the first three patterns. For example, in the term chilliness:

```
#1 chilliness, coolness, nip:  the property of
being moderately cold.
#2 coldness, frigidness, iciness, chilliness: a
lack of affection or enthusiasm.
```

In the fourth pattern, the relation between quality and trait depends on whether the term under the quality subtype refers to *an attribute of something,* or *an attribute of someone*. The first case corresponds to metaphoric polysemy, while the second corresponds to specialization polysemy.

3. Homonymy Patterns: in general, homonymy cannot be considered as a type of regular polysemy. Nevertheless, we cannot exclude the existence of homonymy patterns. WordNet contains few homonymy patterns such as the following pattern: *organism#<animal, plant>*, where we find type mismatch between the subtypes. Specialization or metaphoric polysemy in such patterns is totally excluded.

4. Mixed patterns: this group contains the patterns that were identified to have more than one polysemy type. For example, the pattern attribute#<*quality, trait*> belongs to this group.

In summary, there are some patterns whose sub types indicate type inconsistency. After excluding these patterns, all patterns under the physical *entity* are candidates for specialization polysemy except the patterns under *person*, which contains both polysemy types. In the case of *abstract entity*, most of the patterns under attribute are candidates for metaphoric polysemy. The patterns under *cognition* and *communication* contain both polysemy types, and the rest types are candidates for specialization polysemy.

C. *Polysemy type Assignment*

In this phase, the terms are assigned to the polysemy type of the pattern they belong to. The terms that belong to singleton and mixed patterns are not assigned and they are subject to manual treatment in the validation phase. The terms that belong to specialization polysemy patterns are assigned to the polysemy operation based on the synset synonyms as described in Section IV. Formally, we classify the specialization polysemy cases into three sub classes as follows.

Definition 10 (missing parent synsets sub class) Let $<Ts, s_1, s_2>$ be a specialization polysemy case. Let $Terms_1 = s_1.Terms$, $Terms_2 = s_2.Terms$. The synsets s_1, s_2 are considered to belong to the missing parent sub class, if the following holds: $Terms_1 \setminus (Terms_1 \cap Terms_2) \neq \phi \wedge Terms_2 \setminus (Terms_1 \cap Terms_2) \neq \phi$.

Definition 11 (missing relation synsets sub class) Let $<Ts, s_1, s_2>$ be a specialization polysemy case. Let $Terms_1 = s_1.Terms$, $Terms_2 = s_2.Terms$. The synsets s_1, s_2 are considered to belong to the missing relation synsets sub class, if the following holds: $(Terms_1 \subset Terms_2 \wedge Terms_1 \neq Terms_2) \vee (Terms_2 \subset Terms_1 \wedge Tserm_1 \neq Terms_2)$.

Definition 12 (merge synsets sub class) Let $<Ts, s_1, s_2>$ be a specialization polysemy case. Let $Terms_1 = s_1.Terms$, $Terms_2 = s_2.Terms$. The synsets s_1, s_2 are considered to belong to the merge synsets sub class, if the following holds: $Terms_1 = Terms_2$.

D. *Validation*

In this phase, we manually validate the assigned polysemy type. This phase includes three tasks:

1. **Validation of the assigned polysemy types**: we check whether each of the nouns belong to its assigned polysemy type.
2. **Assigning the polysemy type**: for the terms that belong to the singleton and mixed patterns.
3. **Excluding of false positives**: we exclude the false positives from the terms of the automatic assigned groups. Our judgments during the validation are based on knowledge organization. Word etymology and linguistic relatedness have secondary role.

In Table III, we show the results of our validation for sample patterns. An example for false positives that we found in the common parent group: the meanings of term *apprehender* are homographs.

```
#1 knower, apprehender: a person who knows or
apprehends.
#2 apprehender: a person who seizes or arrests.
```

TABLE III. SAMPLE PATTERNS VALIDATION

# of instances	Pattern	Assigned polysemy Type	# of False positives
2025	Common Parent	Spec. polysemy	93
164	attribute#property,quality	Metaphoric	7
88	attribute#quality,trait	Metaphoric	4
45	vascular plant#herb,woody plant	Spec. polysemy	1

408	event#act,happening	Metonymy	29
328	act#action,activity	Metaphoric	0
21	artifact#commodity,covering	Spec. polysemy	10
56	attribute#property,trait	Metaphoric	0
19	animal#invertebrate,larva	Spec. polysemy	0
26	woody plant#shrub,tree	Spec. polysemy	0

E. Applying the polysemy relations and operations

In this phase, we annotate the resulting metaphoric and homonymy cases explicitly as described in Section IV. For resulting specialization polysemy cases, we apply for each case one of the following three specialization polysemy operations according to the specialization polysemy sub group it belongs to.

1. **Adding missing parent operation:** Let s_1, s_2 be a missing parent case. Let $T_1=\{t_{11},..,t_{1n}\}$, $T_2=\{t_{21},..,t_{2m}\}$ be the synonyms of s_1 and s_2 respectively. Let $T_p=T_1 \cap T_2$, $T_{np1} = T_1 \setminus T_p$, $T_{np2} = T_2 \setminus T_p$. Let t be the preferred term of T_p. Let r be the common root of s_1 and s_2. Let t' be the preferred term in r. We create a common parent S_p of s_1 and s_2 as follows:

i) Create a new synset S_p such that:
The lemmas $S_p = T_p$;
The gloss of S_p = t ." is a ". t'.

ii) The lemmas of $s_1 = T_{np1}$

iii) The lemmas of $s_2 = T_{np2}$

iv) Connect S_p to r via the *is-a* relation

v) Connect s_1 to S_p via the *is-a* relation

vi) Connect s_2 to S_p via the *is-a* relation

vii) Remove redundant relations.

In Figure 4, we show an example for adding a missing parent. In this figure, the hierarchical relations between the two synsets and their hypernym synset in (a) are considered to be redundant and removed in (b).

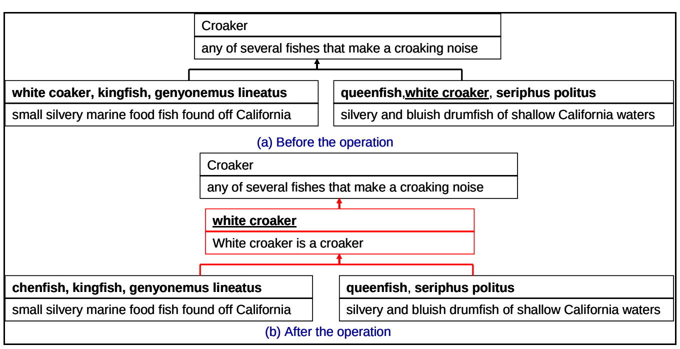

Figure 4. Example for adding a new missing parent

Adding missing relation operation: Let s_1, s_2 be a missing relation case. Let $T_1=\{t_{11},..,t_{1n}\}$, $T_2=\{t_{21},..,t_{2m}\}$ be the synonyms of s_1 and s_2 respectively. Let $T_p=T_1 \cap T_2$, $T_{np1} = T_1 \setminus T_p$, $T_{np2} = T_2 \setminus T_p$. Let s_1 be the synset such that $T_1 \setminus T_p = \phi$.

i) Connect s_1 to s_2 such that s_2 *is_a* s_1.

ii) The lemmas of $s_2 = T_{np2}$.

iii) Remove redundant relations.

In Figure 5, we show an example for adding a missing relation.

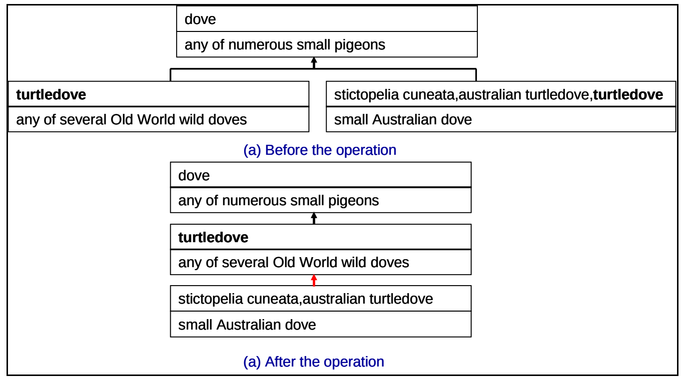

Figure 5. Example for adding missing relation

2. **Merge operation:** Let s_1, s_2 be two synsets of a merge case. We keep the synset with senses rank 1 as follows.

i) The gloss of s_1 = the gloss of s_1 .";". the gloss of s_2.

ii) The relations of s_1 are the union of the relations of both synsets.

iii) Remove redundant relations.

In Figure 6, we show an example for merge operation.

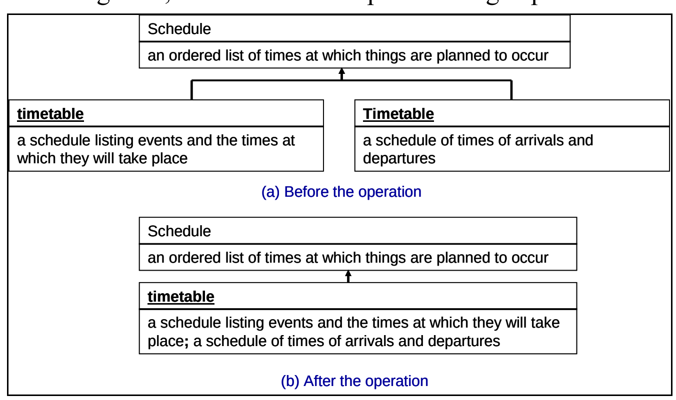

Figure 6. Example for merge operation

VI. OVERLAPPED TERMS SYNSETS

The relation between terms and synsets in WordNet is many to many. This means that it is possible for a term, or a synset, participates in more than one polysemy relation or operation of the same type (e.g., specialization polysemy operation). Considering such cases is very important, since the specialization polysemy operations make changes in the ontological structure and the synset synonyms. Changes in the ontological structure affect the structural patterns, and changes in the synset synonyms affect the criteria for determining the polysemy operations between the synsets in specialization polysemy cases. The relation between specialization polysemy synsets is a binary relation and the specialization polysemy operations are applied pair wise.

To guarantee the correctness of the operations in cases of overlapped terms and synsets, we need rules for structural patterns construction and rules that control the order in which the operations are applied. In the following, we explain the strategy we are using to avoid multiple solutions and enforce the correct organization of such cases. The following figure represents an extreme case of overlapped terms and synsets.

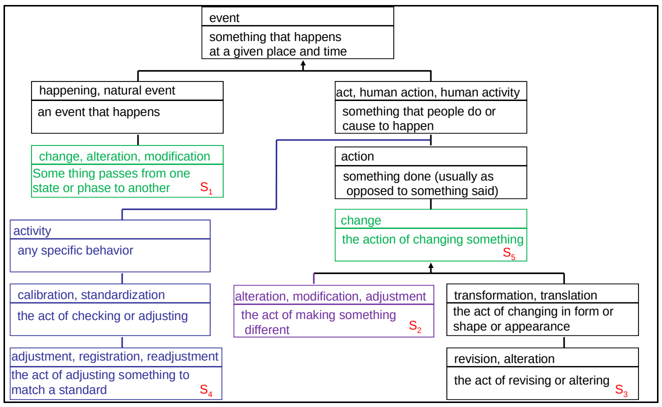

Figure 7. Example for Overlapped terms and synsets

In this example, we can see the following overlapped terms: The terms *alternation* and *modification* are found in s_1 and s_2. The term *alternation* in s_1, s_2, and s_3. At the same time we find the term *change* in s_1 and s_5 and the term *adjustment* in s_2 and s_4. The synset s_2 participates in two operations: a missing parent operation in s_1, s_2 and another missing parent operation in s_2, s_4.

Another important issue in the case of synset overlapping is that it is possible for a synset to follow more than one structural pattern. For example, the synset s_3 may follow the folloiwng patterns:

1. s_1, s_3 belong to the pattern event#< happening, act>
2. s_2, s_3 belong to the common parent *change*
3. s_4, s_3 belong to the pattern act#<action, activity>

To handle this case and similar cases, we propose the following rules:

- **Patterns constructing rule:** construct the patterns bottom up and consider the first possible pattern. Applying this stratgy, s_3 has only one pattern, which is the common parent with s_2. At the same time s_2 belongs to two patterns, because the shared terms in s_2, s_3 and s_2, s_1 are different.
- **Operation Priority rules:** the operations are applied according to the following priority rules:

Synset level rule: apply the operations in a top down manner. For example, following this rule, we apply the operation on s_1, s_5 before the operation on s_1, s_2.

Number of shared Terms rule: order the operations according to the number of shared terms. Following this rule, we apply the operation on s_1, s_2 before the operation on s_2, s_3. The operations on s_2, s_3 and the operation on s_2, s_5 have the same priority.

Resulting changes rule: in case a synset is participating in more than one operation, the type of operation may change according to resulting changes from previous operations. For example, the operation on synset s_2, s_4 is mising parent operation. The result of the operation on s_1, s_2, which is applied before the operation on s_2, s_4 leads to changing the operation on s_2, s_4 from missing parent operation to a missing relation operation.

Through the patterns construction rule, we guarantee two important aspects of specialization polysemy relatedness:

- **Specifity of terms**: the terms whose patterns can be constructed in the lower level are more specific than the terms who have other overlapping patterns in a higher level in the ontology.
- **Meaning relatedness**: through the bottom-up construction and discarding the higher level overlapped patterns, we guarantee that the captured meanings belong to the same, or very near ontology level and thus the meanings are more related than those of the dicarded patterns.

We explain the importance of the operatin priority rules as follows:

- **Synset level rule:** the synsets whose pattern is found at higher level of the ontology are usually a more general meanings. Thus applying the rules in a top level fashion guarantees that the results do not conflict with the orignal ontological structure. For example, it is clear that s_5 is a more general meaning of s_1. Applying the missing parent operation on s_1, s_2 before the missing relation operation on s_1, s_5, makes it impossible to keep the original relatedess between s1 and s_5.
- **Number of shared terms rule:** the ratioanle behind this rule is that synsets that share more terms are more related. We find evidence for this rule in polysemy reduction approaches. In polysemy reduction approaches, they consider synsets that share more than one term to be merged.
- **Resulting changes rule:** this rule is a consequence of the previous rules. We sort the rules according to the synset level rule and number of shared terms rule. These rules guarantee the correctness of the resulting

changes. Since we apply the operations pair wise, each of these operations affects the ontological structure and the terms of the operated synsets. Thus the subsequent operations should be applied on the resulting structure of the previuos operations.

In Figure 8, we show the final result of applying the operations on the synsets in Figure 7.

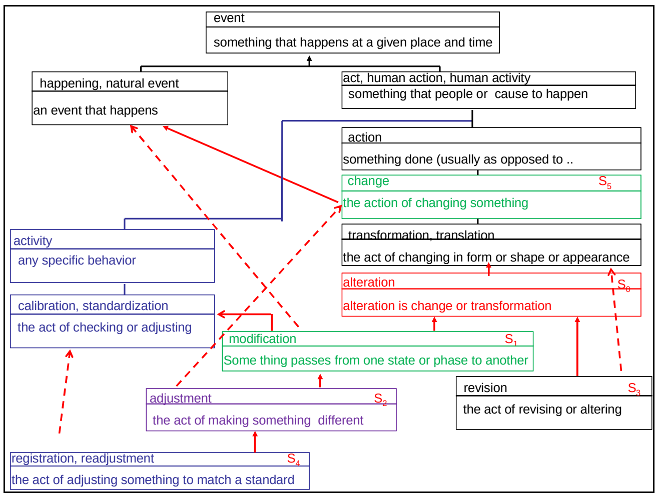

Figure 8. Solving overlapped terms and synsets

In Figure 8, the red colored lines and synsets are newly added. The dashed red lines are the removed relations. We apply the operations in the following order:

1. Missing relation operation on s_1, s_5 (according to the synset level rule). This affects s_1 and s_5 in the following way. We connect s_1 to the synset *hapenning.* The synset s_1 now is a subtype of s_5 and the term *change* is removed from s_1.
2. The operation on the synsets s_1 and s_2 has changed now to a missing relation instead of the original operation missing parent.
3. We apply the missing relation operation on s_1, s_2 (according to the number of shared terms rule) . The synset s_1 is connected to the synset *calibration* and s_2 is connected to s_1. The relation between s_2 and the synset *change* is removed due to the relation redundancy rule. The terms *alteration* and *modification* are removed from s_2.
4. The operation on s_2, s_4 has changed to missing relation instead of the orignal missing parent. There is no change in the operation s_2, s_3.
5. Missing parent operation on s_2, s_3. This leads to creating a new synset s_0. The synset s_0 has the term alteration only. The synsets s_2 and s_3 are connected to s_0. The relation between s_3 and transformation is removed due to redundancy rule.
6. Missing relation operation on s_2, s_4. The term adjustment has been removed from s_4.

VII. RESULTS AND EVALUATION

For the manual validation described in Section V and the evaluation process described in this section, we have developed a special user interface (see Figure 9). This user interface provides the local view of the polysemy cases. For each polysemy case, the user can view also the polysemy type of the displayed polysemy case and the polysemy operation (applicable for specialization polysemy cases). The user can then agree with the suggested polysemy type/ polysemy operation or he can choose one of the provided alternative polysemy types. If the user can not decide, he can choose "No decision".

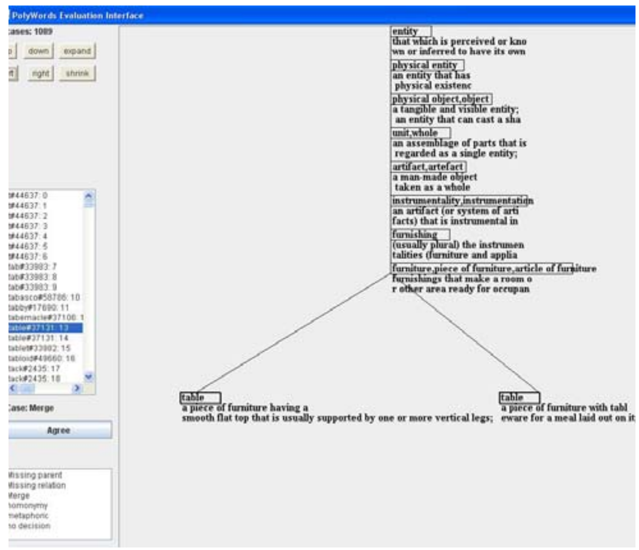

Figure 9. Polysemy Evaluation Interface

In Table IV, we present the results of our approach after the manual validation.

TABLE IV. VALIDATED RESULTS OF THE ALGORITHM

Polysemy type	# of cases	# of cases in percentage (%)
Metaphor	1040	8.2
Homograph	1054	8.3
Spec. Polysemy	10200	80.7
Systematic and Others	361	2.8
Total	12655	100

The cases in the column systematic and others are the cases that we think that they should be processed in a subsequent phase of our approach in the framework of approaching CORELEX systematic polysemy or cases, where the presence of the polysemous term in one of the synsets is inappropriate and should be removed from one of the synsets. An example for such cases is the term *senate* that appears in the synset and its direct hypernym:

United States Senate, U.S. Senate, US Senate, Senate: the upper house of the United States Congress.

=> *senate: assembly possessing high legislative powers*

In Table V, we present the classification of specialization polysemy. The total number of reduced polysemous cases is 10200. The total number of merged synsets represents about

18% of the total processed cases. At the same time we have added 4212 new synsets and 8293 new *is a* relations, while have deleted 1907 synsets and 1907 *is_a* relations. This means that in our approach we have increased knowledge rather than decreasing knowledge to solve the polysemy problem.

TABLE V. SPECIALIZATION POLYSEMY RESULTS

	# of words	# of words in percentage (%)
Missing parent	4212	41.3
Missing relation	4081	40
Merge	1907	18.7

To evaluate our approach, 1020 cases have been evaluated by two evaluators. In Table VI, we report the statistics of the evaluation, where the column polysemy type refers to homonymy, metaphoric, metonymy, or specialization polysemy and the column polysemy operation refers to creating missing parent, adding missing relation, or merging operation. Note that, polysemy operation is applicable in case of specialization polysemy. The table presents the agreement between the evaluators and our approach. The third row represents the number of cases, where at least one evaluator agrees with our approach.

TABLE VI. EVALUATION RESULTS

	Polysemy type agreement	Polysemy operation agreement
Evaluator 1	979 ≈ 96%	924 ≈ 90.5%
Evaluator 2	945 ≈ 92.5%	855 ≈ 84%
Partial agreement	1006 ≈ 98.5%	978 ≈ 96%

As we can see from the results above, although the agreement with the approach is high, in many cases, the evaluators agree on the specialization polysemy type but disagree on the operation type. The explanation for this is that the operation is decided according to the nature of lemmas in both synsets as explained in Section IV.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we introduced a pattern based approach for solving the polysemy problem in WordNet. Our approach deals and covers all polysemy cases at all ontological levels of wordNet. Furthermore, it improves the ontological structure of WordNet by transforming the implicit relations between the polysemous senses at lexical level into explicit semantic relations. The manual treatment in two phases of the approach guarantees the quality of the approach result. Beside the manual evaluation of our approach, we plan to run an indirect evaluation to test the effects of our approach in terms of precision and recall. We applied our approach on all polysemous nouns in WordNet. In a subsequent phase, we are going to extend our algorithm to handle verbs, adjectives and adverbs.

The main contributions of this work are at two levels:
at the conceptual level, we are providing a new foundation towards the problem of polysemy. At the implementation level, we aim to improve the quality of NLP and knowledge-based applications, especially in the field of the semantic search.

REFERENCES

[1] A. A. Freihat, F. Giunchiglia, and B. Dutta, “Approaching Regular Polysemy in WordNet,” in proceedings of the 5th International Conference on Information, Process, and Knowledge Management (eKNOW), 2013, Nice, France, pp. 63-69.

[2] G. A. Miller, “WordNet: a lexical database for English,” Communications of the ACM 38 (11), November 1995, pp. 39 – 41.

[3] J. Gonzalo, “Sense Proximity versus Sense Relations,” Proc. of the Second Global WordNet Conference, Brno, Czech Republic, January 20-23, 2004, pp. 5-6.

[4] P. P. Buitelaar, “CORELEX: Systematic Polysemy and Underspecification,” PhD thesis, Brandeis University, Department of Computer Science, 1998.

[5] R. Mihalcea and D. I. Moldovan, “EZ.WordNet: Principles for Automatic Generation of a Coarse Grained WordNet,” FLAIRS Conference, 2001, pp. 454-458.

[6] L. Barque and F. R. Chaumartin, "Regular Polysemy in WordNet", JLCL, vol. 24, no. 2, 2009, pp. 5-18.

[7] B. Nerlich and D. D. Clarke, “Polysemy and flexibility: introduction and overview,” B. Nerlich, Z. Todd, V. Herman and D. D. Clarke (Hg.), Polysemy: Flexible Patterns of meaning in Mind and Language, Berlin, New York: Mouton de Gruyter, 2003, pp. 3-29.

[8] J. Gonzalo, I. Chugur, and F. Verdejo, “Sense clusters for Information Retrieval: Evidence from Semcor and the EuroWordNet InterLingual Index,” ACL-2000 Workshop on Word Senses and Multi-linguality, Association for Computational Linguistics, pp. 10-18.

[9] J. Pustejovsky, The Generative Lexicon, Cambridge: MIT Press, 1995.

[10] R. Mihalcea, “Turning WordNet into an Information Retrieval Resource: Systematic Polysemy and Conversion to Hierarchical Codes,” IJPRAI, vol. 17, no. 5, 2003, pp. 689-704.

[11] W. Peters, “Detection and Characterization of Figurative Language Use in WordNet,” PhD thesis, Natural Language Processing Group, Department of Computer Science, University of Sheffield, 2004.

[12] R. Navigli, “Word sense disambiguation: a survey,” ACM Comput. Surv., vol. 41, no. 2, 2009.

[13] N. Tomuro, “Systematic Polysemy and Inter-Annotator Disagreement: emirecal Examinations,”

Proc. of the First International Workshop on Generative Approaches to Lexicon, 2001.

[14] M. Palmer, H. T. Dang and C. Fellbaum, “Making fine-grained and coarse-grained sense distinctions, both manually and automatically,” Natural Language Engineering (NLE), vol. 13, no. 2, 2007, pp. 137-163.

[15] N. Verdezoto and L. Vieu, “Towards semi-automatic methods for improving WordNet”, Proc. of the 9th International Conference on Computational Semantics, Oxford, UK, 2011.

[16] J. Apresjan, “Regular Polysemy,” Linguistics, vol. 142, 1974, pp. 5-32.

[17] W. Peters and I. Peters, “Lexicalized systematic polysemy in WordNet,” Language Resources and Evaluation, 2000.

[18] T. Veale, "A Non-Distributional Approach to Polysemy Detection in Wordnet," doi:10.1.1.146.5566.