

Building Verb Predicates: A Computational View

Fernando Gomez

Dept. of Computer Science

University of Central Florida, Orlando, FL 32816

gomez@cs.ucf.edu

Abstract

A method for the definition of verb predicates is proposed. The definition of the predicates is essentially tied to a semantic interpretation algorithm that determines the predicate for the verb, its semantic roles and adjuncts. As predicate definitions are complete, they can be tested by running the algorithm on some sentences and verifying the resolution of the predicate, semantic roles and adjuncts in those sentences. The predicates are defined semiautomatically with the help of a software environment that uses several sections of a corpus to provide feedback for the definition of the predicates, and then for the subsequent testing and refining of the definitions. The method is very flexible in adding a new predicate to a list of already defined predicates for a given verb. The method builds on an existing approach that defines predicates for WordNet verb classes, and that plans to define predicates for every English verb. The definitions of the predicates and the semantic interpretation algorithm are being used to automatically create a corpus of annotated verb predicates, semantic roles and adjuncts.

1 Introduction

This paper deals with the definition of verb predicates which will make possible the determination of verb meaning, semantic roles, adjuncts, and attachment and meaning of post-verbal PPs. Hence, the adequacy of the definitions is measured by comparing the output of a semantic interpretation algorithm with the solution of those semantic interpretation tasks on sentences randomly taken from any corpus, or typed by a user at the console. The algorithm, thus, must provide immediate feedback by testing the definitions on these randomly selected sentences. In (Gomez, 2001), generic predicates have been defined for WordNet 1.6 (henceforth, WN) verb classes (Fellbaum, 1998). The semantic roles of the predicates are linked to the

selectional restrictions (categories in WordNet ontology for nouns) and the grammatical relations that realize them. The selectional restrictions of the predicates are solidly grounded on the WordNet ontology for nouns (Miller, 1998), whose upper level ontology has been modified and rearranged (Gomez, 2004) based on the feedback obtained by testing the predicate definitions.

However, we have found out that the initial idea in that work of defining predicates for a WN verb class which will be also valid for most of the verbs under that class has proven to be too optimistic because many of the verb forms included under that class realize their semantic roles by different selectional restrictions and grammatical relations. This is due to the fact that many of the verbs under a given class have been grouped in many instances by some kind of implication, or *troponymy* (Fellbaum, 1998), rather than by sharing semantic roles in a hierarchy of predicates. Even in many cases, some of the verbs in the same synset list differ semantically and syntactically between them. For instance, the third sense of “gain” in WN is: “profit, gain, benefit (derive benefit from).” The *theme*, the thing obtained, is syntactically realized by the PP [from NP] for two of the verbs listed in that synset “benefit” and “profit,” while the *theme* for “gain” is realized by a direct object. The differences in grammatical relations are even more prevailing within the verbs under one given class. But, in those cases in which the verb polysemy is not high, one predicate definition for the verb class may apply to many of the verb forms under that class. Notwithstanding these problems, WN verb classes have provided an important basis for the construction of a general ontology of predicates that will cover every English verb. Moreover, if one considers that there are 5752 verbs in WordNet 1.6 having only one sense and 2199 verbs having exactly two senses, the pred-

icates that have been constructed for the verb classes of 7,951 verbs are very close to be done. However, the method explained in this paper deviates considerably from the WN approach to constructing verb meaning. In particular, it eschews the synset list in favor of predicate definitions for individual verbs as opposed to a list of synonymous verbs, and allows for an easy integration of a new predicate into a list of already defined predicates for a given verb.

Briefly, the algorithm (Gomez, 2001) that tests the predicates is as follows. For every verb in a sentence, we provide a list of predicates for that verb. These predicates can be viewed as contenders for the meaning of the verb. The goals of the algorithm are to select one predicate from that list, thus determining the sense of the verb, identify its semantic roles, and adjuncts and attach post-verbal PPs. All these tasks are simultaneously achieved. For each grammatical relation (GR) in the clause (starting with the NP complements) and for every predicate in the list of predicates, the algorithm checks if the predicate explains the GR. A predicate *explains* a GR if there is a semantic role in the predicate realized by the grammatical relation and the selectional restrictions of the semantic role subsume the ontological category of the head noun of the grammatical relation. This process is repeated for each GR in the clause and each predicate in the list of predicates for the verb of the clause. Then, the predicate that explains the most GRs is selected as the meaning of the verb. The semantic roles of the predicate have been identified as a result of this process. In case of ties, the predicate that has the greatest number of semantic roles realized is selected. Every grammatical relation that has not been mapped to a semantic role must be an adjunct or an NP modifier. The entries for adjuncts are stored in the root node *action* or *description* (for stative verbs) and are inherited by all predicates in those categories. Adjuncts are identified after the predicate of the verb has been determined because adjuncts are not part of the argument structure of the predicate.

In the next section, we will show how to define predicates for individual verbs as different from WN verb classes, how these predicates for individual verbs can reuse entries in the generic predicates for WordNet verb classes, and how they are integrated into the ontology of predicates that have been defined with the help of WN verb classes. Section 3 provides a discus-

sion of the semiautomatic construction of the predicates. Section 4 gives a view of the upper-level ontology of predicates, section 5 discusses the testing, and sections 6 and 7 related research and conclusions, respectively.

2 Defining Predicates for Individual Verbs

The approach to defining predicates for individual verbs is similar to the way in which the senses for a verb are given in a dictionary. The difference lies in the level of detail provided. For each verb sense, one must provide: a) a predicate and a hierarchy where to insert it, b) the semantic roles for the predicate, and c) the selectional restrictions and grammatical relations for each role. Of course, in the construction of the definitions one must keep in mind the interpretation algorithm because the definitions must determine verb meaning, semantic roles and adjuncts. Except for that, our approach is similar to a dictionary definition in the sense that the predicates for that verb are listed in the same way as the senses of a verb are listed in the dictionary. The lexical definition of the predicates is a frame-like representation containing the selectional restrictions followed by the grammatical relations for that role given those selectional restrictions. The syntax for a semantic role is:

```
(role ( (<slr> (<grs>
          (<slr> (<grs>
                .....
                (<slr> (<grs>))))))
```

Where *< slr >* stands for any number of selectional restrictions, and “*< grs >*” for any number of grammatical relations. The order in which the grammatical relations are listed is irrelevant. However, the order of the selectional restrictions is relevant in the sense that the first selectional restriction that matches is selected and the others in the list are not tried. Hence, the list of selectional restrictions is a preference list (Wilks, 1975). This preference list is critical in selecting the correct sense for many head nouns in the noun phrases of the verb arguments.

A selectional restriction preceded by the sign “-” (-slr) means that the semantic role is not realized by that ontological category. The grammatical relations for PPs are represented by writing “prep” followed by the prepositions that realize the semantic role, e.g. (prep about on

...). Some of the grammatical relations are: subj (the sentence has a subject), subj-if-obj (the sentence has a subject and a direct object), obj (direct object), obj2 (second post-verbal NP), cp (any complement clause), cp-inf (an infinitival complement clause), cp-that-clause (a complement clause introduced by a that clause), cp-ing (a complement clause introduced by gerund), prep-cp (a complement clause introduced by a preposition), etc.

Suppose that one wants to define predicates for the verb “gain.” The first three predicates are:

```
(gain
[gain-weight ;gain9 in WN
(is-a(c-change-state-of-animate))
(agent(human animal)(subj-if-obj))
(theme(weight_unit1 )(obj))]

[gain-reach-a-location;gain4 in WN
(is-a(reach-a-location))
(to-loc(location) (obj))]

[gain-possession;gain8in WN
(is-a(transfer-of-possession))
(theme(possession)(obj))
(from-poss(human-agent)((prep from)))
(source-t(-human-agent thing)
((prep from))(prep-cp by from)))]
```

We have included the WN verb senses following the predicate, but they can be omitted. The reason for putting them is to produce an output that can be checked against WN verb senses (see Section 5). We could have also included the sense numbers of *Longman Dictionary of Contemporary English* (Longman Group, UK Limited), one of the dictionaries that we have consulted in constructing the predicates. The first predicate *gain-weight* is a subpredicate of *c-change-state-of-animate*, cause-change-of-state-of-animate-being, which has been already defined for a class of WN verbs. In fact, one does not need to indicate the *agent* role because it is already indicated in its parent predicate. The only roles that one needs to list are those that differ from those in its immediate super-predicate. The predicate *gain-reach-a-location* is a subpredicate of *reach-a-location*, which has been already defined in principle for all WN verb forms that fall under the synsets “reach1” and “scale4.” The *to-loc* role, which is identical to the parent predicate, does not need to be listed either. Hence, all the work already

done in defining predicates for WN classes can be reused in defining predicates for individual verbs which, because of their high degree of polysemy, cannot be handled by the predicates defined for WN verb classes. The predicate *gain-possession* expresses the sense of the verb when the things gained are money, assets, etc. When the source of the things transferred are humans or social groups (“human-agent” stands for humans or social groups) the semantic role is called *from-poss*, e.g. “She gained much money from the firm.” The role *source-t* is used when the source is not a human-agent, e.g. “He gained much money from the mines or from/by selling the estate.” The preposition that realizes this role is “from.” The prep-cp stands for a complement clause introduced by a preposition(s). In this case, the prepositions “from” (from selling) and “by” (by selling). The -human-agent in the source-t indicates that this role is not realized by the ontological category of *human-agent*. The next two:

```
[gain-something;gain1,gain2,gain3 in WN
(is-a(transfer-of-something))
(theme(thing) (obj))
(from-poss(human-agent)((prep from)))
(source-t(-human-agent thing)
((prep from))(prep-cp by from)))]

[gain-increase;gain6,gain7 in WN
(is-a(increase))
(agent(nil)(nil))
(theme(substance magnitude process
action thing) (obj))
(belonging-to:theme(-human-agent
-animal thing) (subj-if-obj)))]
```

The predicate *gain-something* is the most general, because the selectional restriction of its *theme*, *thing*, is the most general concept in the ontology. This predicate will be also selected by the algorithm as the second choice for the verb “gain” in “She gained much money from the lottery.” The first predicate will be *gain-possession* and the second predicate *gain-something*. If one does not want this to happen and limit the choices just to one predicate, namely *gain-possession*, for this sentence, one needs to rewrite the entry for the theme of *gain-something* as: (**theme (-possession thing) (obj)**). This will exclude the things that have *possession* as a hypernym, or superconcept. The predicate *gain-something* is a default predicate for “gain.” That predicate contains implicitly

many predicates that are unanalyzed, in particular those whose *theme* is a nominal, e.g., “gain control/independence/success, etc.” The meaning of these predicates can be inferred, or generated from the meaning of the nominal.

The predicate *gain-increase* is a subpredicate of *increase*. The entry “(agent(nil)(nil))” means that this predicate does not have an agent, so it will not inherit this role from any superpredicate. The meaning of this predicate expresses those propositions in which entities other than humans or animals gain something. The role *belonging-to:theme* means that the filler of this role is in the relationship of *belonging-to* to the filler of the role *theme*. The relation *belonging-to* is a very general relationship including many other relations such as *at-loc*, *attribute*, and others. This role is also shared by other predicates. Thus, the analysis of the sentence “the pond gained water” is *something caused the water belonging-to the pond to increase*. In this sentence, the *belonging-to* relation stands for *at-loc*, something that can be inferred by additional analysis of the ontology of the arguments of the relation. A similar analysis is produced for “the object gained speed uniformly.” Both sentences can be rephrased by saying “the water in the pond increased,” and “the speed of the object increased.”

Order of the Predicates

As indicated, the order of the predicates is relevant in the sense that, in case of ties (two or more predicates realize all their semantic roles), they will be preferred as the predicate for that verb in the order in which they are defined. This order has nothing to do with frequency of senses for a given verb, but rather with the generality of the ontological categories in the selectional restrictions of the semantic roles. For instance, had the predicate *gain-weight* been defined after *gain-something*, *gain-weight* would have been ranked second after *gain-something* as the predicate of “gain” in the sentence “The teacher gained weight.” If the ontological category of the selectional restriction of at least one role, says, r_1 , in predicate, say, p_i , is not in the same path as the selectional restriction of role r_1 in predicate p_j , then it makes no difference in which order the predicates are defined. That is the case for *gain-weight* and *gain-reach-a-location*, because the ontological categories of their *themes* are not in the same path.

The definition of the predicates is provided without any knowledge about what could be

the meaning of the nouns in the sentence. The goal is that the definition of the predicates help determine the senses of the nouns as much as possible. For instance, if one types the sentence “the kite flew into the sky,” the system selects two predicates for “fly”: the first one: *fly* which is a subpredicate of *change-location*, (an animate being changes location from a place to another). This is so because one of the senses of “kite” in WN is a hawk(kite2). Thus, kite2 is selected as the *agent* of *fly*. The second predicate is *cause-to-fly* (a subpredicate of *cause-to-change-location*), in which “kite” is taken as plaything, a toy, (kite1). In this case, “kite1” is the *theme* and the *agent* or *inanimate-cause* are unknown. During one of our demos, somebody typed the sentence “She left the keys in her car,” and the system came up with the following two predicates for “left”: *leave-a-place* and *leave-behind*. Our first reaction was that something had gone very wrong in selecting the predicate *leave-a-place*. But upon looking into it, we found out that one of the senses for “keys” in WN is the “Florida keys,” a location. Hence, the interpretation of the system for “left” was correct, given that sense for “keys”: “somebody left a place (the Keys) using her car as the instrument for the change of location.

3 A software Environment for the Semiautomatic Definition of Predicates

We have developed a software environment that has the following components: a parser, the semantic interpreter that uses the predicates to interpret the sentence, a corpus (*The World Book Encyclopedia*, World Book, Inc. Chicago) divided into different sections for defining, refining and testing the predicate definitions, a skimmer that searches for sentences in the corpus containing the verb for which predicates are being defined, and a mechanism for dragging in sentences from the corpus into the system for testing. An interface has been also implemented for those who abhor Lisp parentheses. The interface interactively asks for the semantic roles, the selectional restrictions and the grammatical relations for a given predicate. Suppose that one wants to define predicates for the three senses of the verb “confuse” provided by *Longman*, namely “to cause to be mixed up in the mind,” “to mix up in one’s mind (with)” and “to put in disorder, make less clear or more difficult to deal with ..”

```
p1: [cause-to-confuse-perplex
(is-a(c-change-state-of-animate))
(agent(human-agent animal)
      (subj-if-obj))
(theme(human-agent animal)(obj))
(inanimate-cause(thing)
      (subj-if-obj})]
```

```
p2:
[confuse-something-with-something
(is-a(misunderstand))
(agent(human-agent animal)
      (subj-if-obj))
(theme (thing) (obj))
(co-theme(thing)((prep with)))
(require(co-theme))]
```

```
P3:
[confuse-issue-muddy
(is-a (obscure-ideas-information))
(agent(human-agent animal)
      (subj-if-obj))
(theme(-physical-thing
       communication thing) (obj))]
```

The first predicate is straightforward. The *inanimate-cause* is needed because anything in principle can cause to confuse a living thing, e.g., “The light confused the bird and was trapped,” or “These questions confuse even the experts.” The second predicate has no restrictions in the roles *theme* and *co-theme*. Anything can in principle be confused with anything else. The *require* slot can be used in any predicate to indicate that in order for that predicate to be selected by the algorithm as the meaning of the verb, the role expressed in the *require* slot must have been filled by the algorithm. This is not absolutely needed in this case, but it minimizes the number of predicates selected for a verb. Note that, if one removes the *require* slot from p2, and one types the sentence “He confused the teacher,” predicates p1 and p2 will be selected because both predicates will explain the two grammatical relations in the sentence. However, if one types “She confused the crane with a heron,” only p2 will be selected because p2 will explain the PP [with a heron] and p1 will not be able to. The selectional restriction of the *theme* for predicate p3 is anything that is not a *physical-thing* in the ontology, preferring those concepts having *communication* as a hypernym.

There are two ways in which one can proceed in the definition of these predicates. One way is to provide the definitions given above using the interface, which reduces to responding to some commands/questions by the interface,

namely “enter role,” “enter selectional restrictions,” “enter grammatical relations for that role,” “done with this role ?” etc. Then, once the definitions are completed one can skim a section of the corpus for sentences containing the verb for which predicates are being defined and try them (parse and interpret them) in the system in order to test the definitions provided. Most sentences are parsed and interpreted in few seconds.

Another way is to skim the corpus for sentences containing the verb for which predicates are being defined, provide tentative definitions based on the sentences retrieved by the skimmer, and then refine and test the definitions by searching the section of the corpus for refining predicates, and loop as needed. The interface provides with all kinds of functions for searching the noun ontology, which facilitates greatly the task of defining the predicates.

The placing of a predicate in the hierarchy of predicates is a semiautomatic process. The interface has a function that displays all the predicates already built for all the WN senses for a given verb. There are also functions that display the content of the predicates as well as all their superpredicates and subpredicates. This makes the insertion a rather simple task.

Adding a New Predicate

Suppose that one wants to add a new predicate to a list of predicates for a given verb which one thought it was complete. One may discover a new sense that WN or the dictionary has missed, something not uncommon. This is as simple as adding the new predicate definition to the list of predicates for that verb. Suppose that one wants to add the predicate *gain-a-reputation* to the list of predicates for the verb “gain,” because the verb “gain” is frequently used with this concept. This predicate is clearly a refinement of the predicate *gain-something*. Hence it needs to be inserted before it. Prior to introducing the new predicate one may want to check if there is a predicate whose *theme* contains the selectional restriction of “reputation,” or one of its hypernyms. This is a task fully automated. One needs only to type: (find-predicates (theme reputation1)). This will retrieve all predicates whose *theme* has *reputation1*, or one of its hypernyms as its selectional restriction. In this case, the predicate *establish-a-reputation* that has standing1 as the selectional restriction for the *theme* will be retrieved. So, one needs only to insert the predicate *gain-*

a-reputation immediately before *gain-something* as follows:

```
(gain-a-reputation
 (is-a(establish-a-reputation)))
```

4 A Panorama of the Upper-level Ontology of Predicates

We have defined 3,017 predicates. Some of them are very complex, while others contain just two or three semantic roles because other roles are inherited from their superpredicates. The major classes and some of their subclasses are briefly described. The first class is **cause-change-of-state** with 609 subpredicates. This class contains major subclasses with very little relation to each other. Some major subclasses (the number of predicates under each class are given in parentheses following the class) are *cause-change-of-state-of-animate-being* (140 subpredicates) which in turn has subclasses such as *arouse-feelings-emotions* (52) *cause-to-act* (19), *injure-hurt-somebody-or-oneself* (18). Other major subclasses of *cause-change-of-state* are: *increase* (31), *improve* (19), *worsen* (10), *terminate* (16), *physical-change-of-state* (14) (e.g., *solidify*, *liquefy*, etc.) *cause-change-of-integrity* (22), *transform* (9) and others.

The next predicate class is **cause-to-change-location** with 379 predicates. The primary event expressed by this predicate is a cause of change of location of something other than the agent; although the agent may have also changed location. In “Linda carried the books to the library” the agent also has been moved, but the primary event is the fact that Linda caused a change of location of the books. In a sentence such as “The moon circles the earth,” “the moon” is the *theme* and the *agent* or *inanimate-cause* is unknown. The predicate *cause-to-change-location* is not to be confused with *change-location* (see next class) in which the agent and the theme are the same animate being, e.g., “Kelly went to the store.” The major subclasses of **cause-to-change-location** are: *put* (75), *remove* (53), *transport* (23), *propel* (20), *connect* (22), *flow* (12), and *pull*, *push* and *send* with 9 predicates each.

The next class is **change-location** with 238 predicates. Major subclasses are: *walk* (14) (some subpredicates of *walk* are *hike*, *march*, *sneak-walk*, and others), *enter* (10), *leave-a-place* (11), *travel*, *arrive-to-a-place* (10) and others.

The next class is **interact** with 372 predicates. Within this class, the major subclasses are: *communicate* (243), *treat-an-animal-or-human* (26), whose major subpredicates are *treat-unjustly-somebody*, and *behave* (9), *join-a-group-or-a-human* (9) and others.

The next class is **transfer-of-something** with 293. Its major class is *transfer-of-possession* (231), which in turn contains such subpredicates as *give* (31), *get* (132) and others. The next class is **make-or-create-something** with 144 predicates. Some major subclasses are *initiate-something* (30), *create-art* (28), *produce* (10), *prepare-something* (10) and others. Another major class is **judge** with 182 predicates. Some of its subpredicates are *pass-a-negative-judgment-on* (36) (e.g., *disapprove*, *oppose-ideas*, *oppose-people*, and others.), *put-value-to-something* (12), *praise* (7), *accept-admit-a-fact* (30), *confirm-corroborate* (11), *deny-something-to-somebody* (14), *accuse* (11) and others. The next class is **experience** with 110 predicates. Some of its subclasses are: *feel-in-a-state-or-emotions* (25), *perceive* (21), *like-something* (36), *experience-event-state-abstraction* (11) and others. The next class is **think** with 115 predicates. Some of its major subclasses are *analyze* (31), *plan* (13), *reason-conclude* (9), *associate* (7) and others.

The next class is **decide** with 92 predicates. Its major subclass is *exert-control-over* (65), which in turn contains major subclasses such as *restrain*(22) and *manage* (23). Another class is **touch** with 55 predicates. Its major subclasses are *handle-operate* (18) and *hit-something* (24). Another major class with 52 predicates is **spend-something**, which has *invest* (31) as one of its main subclasses.

Some other predicate classes are: **prevent** (45), **move-body-position** (33), **know** (31), **fail-to-do-something** (33), **fight** (31), **expel-substance-from-the-body** (23), **do-act** (27), **appoint-somebody** (15), **allow-something** (21), **support-something** (24), **succeed** (23), and **utilize** (11). To these we need to add 223 predicates for stative verb senses (e.g. *be-at-a-place*, *include*, etc.)

5 Testing

We have tested the predicates in 500 sentences taken from the *The World Book Encyclopedia*. The manually corrected output of the semantic interpreter on these 500 sentences has been made public on November 26, 03 on the

Table 1: Statistics for the Sentences

Predicate for the Verb	87%
Semantic Roles:	91%
Adjuncts:	82%
Post Verbal PP attachment:	89%

authors' homepage (www.cs.ucf.edu/~goomez). These sentences and their subclauses contain close to 400 distinct verbs. Many of these verbs appear in the 500 sentence corpus several times with different senses, while others appear just one or two times. Space limitations prevent us from listing the verbs in this paper. Table 1 summarizes the results of the semantic interpreter on these 500 sentences. We counted as a correct predicate for the verb only the *first* predicate selected by the algorithm. Semantic roles were computed only for those predicates which were correctly selected by the algorithm. As one can see, the results are very good, and, as a consequence, the manual correction of the errors in the sentences takes very little time. In fact, we have made public another set of 500 sentences on March 24, 2004. This new set of sentences tests many verbs that did not appear in the initial corpus, and which exhibit many different senses. The preliminary results on this new set are comparable to the results for the first set. The selectional restrictions are aimed at capturing the possible noun categories for a given predicate, and are not biased to any corpus. The main reasons for failing to assign meaning to a constituent are: over-specification or over-generalization of the selectional restrictions, missing a grammatical relation, sense not contemplated in the predicate definitions for that verb, and insufficient information in the sentence to unequivocally determine the predicate of the verb. The semantic interpreter has become an excellent tool to automatically build a corpus of annotated predicates, semantic roles and adjuncts, requiring very little human effort in validating and correcting the final output. Our additions and restructuring to the upper level noun ontology together with the preference list in the selectional restrictions is making possible for the interpreter to resolve many noun senses, which otherwise would have remained unresolved. Figure 1 depicts the output of the interpreter for one of the sentence in the initial corpus. The interpretation of the sentences in the corpus include relativization,

```
(Clause CL364
(SUBJ: ((ADJ MANY) (ADJ WHITE) (NOUN TRADERS))
(MERCHANT1 TRADER1) <(AGENT)>
)
(ADVERB: ((ADVERB LATER)) LATER
)
(VERB: OPENED <OPEN-AN-ENTERPRISE:(OPEN2)
supported by 2 SRs>
)
(OBJ: ((NOUN POSTS)) (POSITION6 POST3)
<(THEME)>)

(PREP : ON
(PREP-NP: ((PN INDIAN) (NOUN RESERVATIONS))
(RESERVATION1 INDIAN_RESERVATION1)
<(AT-LOC)>)
Attach: Verb Confidence: WEAK)
)
AND
(Clause CL360
(SUBJ: ((ADJ MANY) (ADJ WHITE) (NOUN TRADERS))
(MERCHANT1 TRADER1) <(AGENT)>
)
(ADVERB: ((ADVERB LATER))
LATER
)
(VERB: CHARGED <CHARGE-BILL:(CHARGE3)
supported by 2 SRs>)
(OBJ : ((ADJ UNFAIR) (NOUN PRICES))
(POSSESSION PRICE1 PRICE6) <(THEME)>
)
)
```

Figure 1: Interpreter output for *Many white traders later opened posts on Indian reservations and charged unfair prices*

complement phrases, coordination, subordination and adverbial clauses. The grammatical relation is listed first followed by the semantic role for that grammatical relation in the form $\langle (semantic - role) \rangle$. Hence, the pair $\langle \rangle$ (placed to the right of the grammatical relation) identifies semantic roles. The program prints the predicate followed by the WN verb senses and the phrase “supported by ‘a-number’ SRs,” (where ‘a-number’ stands for a cardinal number), which indicates the number of grammatical relations (SRs) that have realized semantic roles. If the output for that sentence has more semantic roles than the number listed in the phrase “supported by,” those semantic roles are adjuncts. Prepositions can be attached to the verb with a confidence of *strong* or *weak*. All post-verbal PPs deemed to be arguments of the verb are attached as *strong*, while all PPs that are deemed adjuncts can be attached as *strong*

or *weak*. The WN sense number for the verb is printed in parentheses immediately following the predicate. Sometimes more than one WN verb sense is given for a predicate. For instance, for the predicate *travel-change-location* given for the verb “travel” in the sentence “She traveled to Spain,” the following WN senses of “travel” are listed (travel1 travel4 travel5). This means that all WN senses of “travel” in that list are coalesced into the predicate *travel-change-location* (Gomez, 2001). Hence, all these WN senses of the verb have the same meaning. The WN noun senses follow the noun in parentheses. In some instances, the algorithm may provide more than one sense for a noun. To the contrary to verb senses, this list is a ranked list in which the first noun sense listed by the algorithm is the preferred one. Besides this output, the system lists all the superpredicates of the predicates selected for the verbs in the sentence. Thus, *open-an-enterprise is-a initiate-an-enterprise-project is-a organize is-a initiate-something is-a give-rise-to-something is-a make-or-create-something*, and *charge-bill is-a transfer-of-possession is-a transfer-of-something is-a action*.

6 Related Research

This work has benefited from several sources. In linguistics, the work reported in (Pritchett, 1992; Grimshaw, 1990; Pinker, 1989) have been influential. In computational linguistics, the VerbNet project (Dang et al., 1998) and Framenet (www.icsi.berkeley.edu/framenet) (Fillmore et al., 2003) bear relation to this work. Major differences are that our project aims at defining predicates for every English verb in a systematic manner, linking the selectional restrictions for their semantic roles to a well established and widely used ontology for nouns, namely WordNet, and placing them in a hierarchy of predicates where inferences and semantic roles can be inherited. Moreover, the definition of our predicates can be tested by running the semantic interpretation algorithm on any corpus or on sentences typed by the user at the console.

7 Conclusions

An approach to building verb predicates has been presented. The construction of the predicates is essentially linked to an algorithm for determining the semantic roles of the predicates. The definitions of the predicates are done semi-

automatically and are being refined and tested with the help of a semantic interpretation algorithm that uses the definitions for determining verb meaning and semantic roles for sentences selected from a corpus, or entered at the terminal. The algorithm and the predicates are being used to automatically construct a corpus of semantic annotated sentences.

References

- Hoan Trang Dang, Karin Kipper, Martha Palmer, and Joseph Rosenzweig. 1998. Investigating regular sense extensions based on intersective levin classes. In *COLING-ACL*, pages 293–299, Montreal, Quebec.
- C. Fellbaum. 1998. A semantic network of english verbs. In C. Fellbaum, editor, *WordNet: An electronic Lexical Database and some of its applications*, page chapter 3. MIT Press, 1998, Cambridge, Mass.
- C. J. Fillmore, C. R. Johnson, and M. R.L. Petruck. 2003. Background to framenet. *International Journal of Lexicography*, 16:235–250.
- F. Gomez. 2001. An algorithm for aspects of semantic interpretation using an enhanced wordnet. In *Proceedings of the 2nd North American Meeting of the North American Association for Computational Linguistics, NAACL-2001*, pages 87–94.
- F. Gomez. 2004. Grounding the ontology on the semantic interpretation algorithm. In *Proceedings of the Second International WordNet Conference*, pages 124–129, Masaryk University, Brno.
- J. Grimshaw. 1990. *Argument Structure*. MIT Press, Cambridge, Mass.
- George Miller. 1998. Nouns in wordnet. In C. Fellbaum, editor, *WordNet: An electronic Lexical Database and some of its applications*, page chapter 1. MIT Press, 1998, Cambridge, Mass.
- S. Pinker. 1989. *Learnability and Cognition*. MIT Press, Cambridge, Mass.
- B. L. Pritchett. 1992. *Grammatical Competence and Parsing Performance*. The University of Chicago Press, Chicago, Illinois.
- Y.A. Wilks. 1975. Preference semantics. In E.L. Keenan, editor, *Formal Semantics of Natural Language*. Cambridge University Press, Cambridge, UK.