

Using selectional preferences for extracting verb meaning, semantic roles and adjuncts from constituent-based parses

Fernando Gomez

Department of EECS
University of Central Florida
Orlando, FL 32861
CS-TR-12-02 January 24 2012

Abstract

Algorithms are described for the identification of verb predicates, or verb meaning, semantic roles and adjuncts in unannotated corpora. Sentences are parsed using a constituent-based parser. For every main verb on the parse tree, a minimal clause structure is built. The initial clauses are refined by filling null elements using verb subcategorization and verb semantics. Finally, the semantic roles, adjuncts and verb meaning are computed. The algorithms have been tested on the fourth release of Ontonotes and on Wikipedia.

Introduction

The task of defining selectional preferences for unrestricted domains is an undaunted one because of two major reasons a) a semantic interpreter that will work with constituent based parses, and b) a general ontology for the selectional preferences that could be applied across domains. Statistical parsers are very robust, but provide an impoverished parse tree in which long distance dependency and missing elements on the tree are not indicated, let alone filled with their referents. Linguists refer to these missing elements, or null elements, as empty categories. The recovery of these missing elements is a necessary condition for extracting semantic information from parse trees. Empty categories may result from diverse linguistic phenomena, i.e. relativization, missing complements, etc. A semantic interpreter requires that the empty nodes be recognized, and some of them filled with their antecedents. Most of the work to resolve these issues has been based on supervised learning algorithms using annotated corpora. A notable exception is Campbell's work (Campbell 2004) which uses principles of Government and Binding (Chomsky 1981) to detect and recover empty categories. His system outperforms previously learning-based algorithms on the detection and recovery of empty categories. Campbell's algorithm does not use lexical information, nor verb semantics. However, the algorithms explained in this paper are also rule-based algorithms that use general linguistic principles, but also verb lexical subcategorization and verb semantics, and they go beyond Campbell's algorithms in that they also identify verb arguments, verb meaning and adjuncts.

Even if all empty nodes on the parse tree are filled, the problems of deciding which constituents are arguments and which ones are adjuncts remain. In the parse trees for "She

knew when John left" (*SI (S (NP (PRP She)) (VP (VBD knew) (SBAR (WHADVP (WRB when)) (S (NP (NNP John)) (VP (VBD left))))))*) and "She ate when John left" (*SI (S (NP (PRP She)) (VP (VBD ate) (SBAR (WHADVP (WRB when)) (S (NP (NNP John)) (VP (VBD left))))))*) all nodes are filled. However, in the sentence for "knew" the SBAR clause is an argument, but in the sentence for "ate" is an adjunct. The machine learning approaches to role labeling rely on syntactic features on the parse trees to determine the roles and label them (Gildea and Jurafsky 2002). A problem with path features is that the paths on the parse tree vary depending on the sentence (Vickrey and Koller 2008). Moreover, the semantics of the nouns and the verbs is critical to determine semantic roles and verb meaning. Consider the sentence "Joshua is bitten by a tarantula and is driven into a frenzy/hotel" in which the meaning of "drive" and the semantic role of the PP "into" vary depending just on the head noun of the PP ("frenzy" or "hotel.")

In (Gomez 2001; 2004), we have shown how to define verb predicates (Grimshaw 1990) linked to Wordnet verb senses (Fellbaum 1998) and the selectional preferences in their semantic roles are WordNet noun ontology (Miller 1998) categories. In addition, he has described an algorithm that determines verb meaning and semantic roles, using the verb predicates. For every non-auxiliary verb in a sentence, a list of verb predicates for that verb is provided. The goals of the algorithm are to select one verb predicate from that list, thus determining the sense of the verb, and to identify its semantic roles and adjuncts. All these tasks are simultaneously achieved. For each grammatical relation (GR) in the clause and for every verb 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 preferences 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. 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 nodes *action* and *description*

```

[leave-a-place
(is-a(change-location))(wn-map(leave1))
(agent(human animall)(subj))
(from-loc(location)(obj))]
| [leave-give
(is-a(give))(wn-map(leave12))
(agent(human-agent)(subj-if-obj))
(theme(possession thing)(obj))
(to-poss(human-agent)(obj-if-obj2)
(human-agent animall)((prep to)))
(thematic-rule(require(to-poss)))]

```

Figure 1: Two brief definitions for two predicates for the verb “leave”

(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 (Grimshaw 1990).

Figure 1 describes two predicates of the verb “leave.” The *wn-map* slot indicates its corresponding WordNet senses. Not all semantic roles are listed because many of them are inherited from its super-predicates. The role *to-poss* for *leave-give* is realized by an *obj* if followed by an *obj2* (e.g., “She left her daughter a fortune.”) or by the preposition “to.” The entry “thematic-rule” means that that predicate requires the role *to-poss*. Consider the sentence “The farmer left the land.” Both predicates will explain the subject “farmer,” (a farmer is a human - *human-agent* subsumes both humans and organizations) and also the object of the sentence. Other predicates for “leave” will also explain both grammatical relations. For instance *leave3* (*cause to be in a specified state*) will also explain both grammatical relations (e.g., “The farmer left the land in ruins.”). Suppose that the sentence is “The farmer left the land on his tractor.” In this case, the predicate *leave-a-place* will explain “on a tractor,” the *instrument* of *change-location* inherited by *leave-a-place* scoring more roles than the other predicates. If the sentence is “The farmer left the land to her daughter,” the predicate “leave-give” will explain “to her daughter,” resulting in being selected as the predicate for the verb. Thus verb meaning is a side effect of identifying semantic roles. Figure 2 describes the semantic interpreter’s output for the clauses of “urge” and “sail” in the sentence “Achilles urged the Greeks to sail home as he was planning to do.” After the interpretation of the clause, the program prints the hierarchy of verb predicates for the verb predicate selected. It also prints any other verb predicate (if any) that is tied for the meaning of the verb with the one selected. The interpreted clauses can be integrated back into the parse tree if desired.

One problem with this algorithm is that it relies on the parser to determine the complements of the verb. Thus, the parser will indicate that the SBAR clause in “She knew when John left” is a complement of “knew”, but in “She ate when John left” the SBAR clause is a temporal adjunct. In addition, the algorithm requires from the parser to fill the null elements on the parse trees. These are two mayor requirements that statistical parsers do not produce. In addition, the algorithm has not been tested against corpora annotated with semantic roles and verb meaning, and not all WordNet verb senses had been mapped to verb predicates, which, in the present version, are. In the next sections, we explain the clause reconstruction algorithm, rules for determining

clausal complement and choosing between subjects. We end with sections on testing, related work and conclusions.

Minimal Clausal Reconstruction (MCR) from Constitution-Based Parsed Trees

We approach the analysis of parse trees from the point of view that a sentence contains one or more clauses, and each clause has its main verb, arguments and/or adjuncts. The approach bears certain similarity to Winograd’s (Winograd 1983) analysis of systemic grammars for parsing.

The algorithm has four steps, each step receiving as input the output of the previous one. In step1, sentences are parsed using Charniak’s parser (Charniak 2000). In step2, a minimal clausal reconstruction for each non-auxiliary verb in the parse tree is produced. In step3, empty complements are filled. In step4 the semantic interpreter (henceforth, SI) identifies the verb predicates, semantic roles and adjuncts. The algorithms reconstruct the clauses of the sentence from the parse tree in a minimal way (not related to minimalism), because they do not decide which constituents of the clause are arguments, and which ones are adjuncts, and build a list of possible subjects for some clauses when the parse tree has an empty node for the subject position. However, the algorithms do resolve long distance dependency resulting from relativization.

An extended description of the MCR algorithm can be found in (Millward and Gomez 2010). The major tasks are: 1) to recover the verb sequence, indicating whether the sentence is passive or active, 2) to recover all postverbal constituents of the VP (all those constituents dominated by the VP), 3) fill subject gaps, 4) break up attachment of postverbal PPs to NPs, and 5) resolve long distance dependency resulting from relativization. The postverbal constituents are enumerated from left to right as *obj*, *obj2*, *obji*. In a parse tree, these are all constituents dominated by the VP node. These constituents can be NPs, VPs, SBARs or Ss. Consider the parse tree for the sentence “She told the children that Mary left.”

```

(s1 (s (np (prp she)) (vp (vbd told) (np (dt the) (nns children)) (sbar (in that) (s (np (nnp mary)) (vp (vbd left))))))))

```

The MCR algorithm builds:

```

(g864 (subj ((pron she)) verb ((main-verb tell told) (tense vbd)) obj ((dt the) (noun children)) obj2 ((gensym g865 that-clause that)))

```

```

g865 (subj ((pn mary)) verb ((main-verb leave left) (tense vbd)) type (that-clause that) parent-verb (tell g864))

```

Note that the MCR is not saying that the clause for “tell”

```
(g925 (subj ((noun achilles)) ((mythical_being1 achilles1)) (agent))
(verb urged ((main-verb urge urged)) urge-someone (urge1) supported by 3 srs)
obj ((dt the) (pn greeks)) ((european1 greek2)) (recipient))
(obj2 (gensym g926 infinitive) (theme)))
```

urge-someone > advise > trans-infor > communicate > interact > action

```
(g926 (subj ((dt the) (pn greeks)) ((european1 greek2)) (agent))
(verb sail ((main-verb sail sail)) sail (sail1) supported by 2 srs)
(obj ((noun home)) ((residence1 home1)) (to-loc)))
```

sail > travel-by-boat > change-location > action

Figure 2: SI output for the clauses of “urge” and “sail” in the sentence “Achilles urged the Greeks to sail home as he was planning to do.”

has two postverbal complements/arguments as it happens in this case, but rather that it has two constituents. It is up to the SI (semantic interpreter) to decide if they are actually arguments or adjuncts.

The other main task of the MCR is to recover subjects but in a minimal way. If the subject of a clause is missing on the parse, the MCR algorithm chooses the subj and obj of the main clause *in that order* as possible subjects. The MCR builds a Subj entry in the structure and inserts the possible subjects in it, first the subject of the main followed by the object of main. For the sentence “She bought a book of history to learn the truth,” the parse tree is: (*s1 (s (np (prp she)) (vp (vbd bought) (np (np (dt a) (nn book)) (pp (in of) (np (nn history)))))) (s (vp (to to) (vp (vb learn) (np (dt the) (nn truth))))))*)

Which the MCR transforms into:

```
(g880 (subj ((pron she)) verb ((main-verb buy bought)
(tense vbd)) obj ((dt a) (noun book)) prep (of ((noun his-
tory))) obj2 ((gensym g881 infinitive)))
```

```
g881 (subj (((pron she)) (((dt a) (noun book)) prep (of
((noun history)))))) verb ((main-verb learn learn) (tense vb)
type (infinitive) parent-verb (buy g880) obj ((dt the) (noun
truth))))
```

The obj2 in clause G880 is not an argument of “bought,” and “she” is the subject of “learn.” The next step will decide about this. For “She promised/asked Mary to read a book,” the MCR builds the same two possible subjects for “read.” Again, the next step will choose the subject of “read” depending on the subcategorization for “promise” and “ask.” For VP infinitives dominated by NPs, the MCR builds *np-dominated-infinitive*. The SI will have its say in deciding if, in fact, a noun in the NP subcategorizes the infinitival clause.

If the main clause follows a subordinate clause without a subject as it happens in reduced adverbial clauses in front of sentences (e.g., “While visiting Iowa, she bought a farm.”), the subject assigned to the subordinate clause is the subject of the main clause. The MCR builds the verb sequence, identifying passive clauses. If the clause is passive, it does not change the subj to obj. But if the verb has a postverbal constituent, the MCR names it obj2. Hence, for the sentence “He was told the truth,” the MCR builds

```
g863 (subj ((pron he)) verb ((aux (was)) (main-verb
```

```
tell told) (tense vbn) (voice passive)) obj2 ((dt the) (noun
truth))))
```

In recovering subjects of clauses dominated by passive clauses, the MCR builds *unknown-agent* as the subject of the embedded clauses. Thus, for the sentence “He was told to get a loan,” the possible subjects for “get” are *unknown-agent* and “he,” in that order. For “The houses were bought to be sold,” the possible subjects of “sold” are *unknown-agent* and “houses” in that order. If the passive clause has also a [by NP] constituent, the MCR builds it as a possible subject for the embedded clause.

Another major aspect of the MCR algorithm is that it breaks up postverbal prepositional phrases attached to NPs by the parser, except under apposition and coordination. Thus, for the sentences “He rid the city of rats,” “She made a statue of marble,” or “Farmers enrich the soil with fertilizers,” the MCR algorithm will de-attach “of rats,” “of marble,” and “with fertilizers” if they were attached to the NP by the parser. We want the argument structure of the verb predicate to have the first say in attaching postverbal prepositional phrases. This is in agreement with the ideas expressed in (Pritchett 1992) and with minimal commitment models (Marcus 1987; Gorrell 1991; Weinberg 1993). If the SI attaches the PPs to the verb *strongly*, the parser’s attachment is overridden. However, if the SI attaches the PP *weakly*, the parser’s attachment is preferred. Obviously, all verb arguments realized by prepositions are *strongly* attached to the verb, while adjuncts are attached *weakly*. Adjective phrases (ADJP) are mapped into PRED.

Refining the output of the MCR

The major tasks in the algorithms that refine the output of the MCR are explained next. For each clause produced by the MCR, starting with the first clause outputted, do: 1) recognize clausal complements 2) select from potential subjects, 3) activate the SI with that clause, 4) if the SI marks an OBJi as a spurious (see below) delete that OBJi from that clause and run the SI with that clause again. Another task which is critical is to verify if the sentence is actually passive/active by using verb subcategorization and the clause structures built for the sentence.

When the nodes dominated by the VP are clauses, the output of the MCR needs to be cleaned using the verb subcategorization information, prior to activating the SI. The critical entries are the nodes *obj* and *obj2*, which, when used in the verb predicate entries, stand for complements. However, in the output produced by the MCR these entries may stand for subordinate clauses not subcategorized by the main verb. Potential NP complements are left to the SI to decide if they are actually arguments, or adjuncts. If the *OBJi* refers to a clause, the function *Process OBJi That Stand for Clauses* is activated for each *OBJi* in the structure from left to right.

If the algorithm determines that an *OBJi* is a CP (complement phrase), it is left in the structure, else the *Obj* is erased from the structure, and the index of any *OBJi* still in the structure is replaced with *OBJ(i-1)*.

These are the main rules in the function for deciding if an *OBJi* standing for a clause is a CP (a complement phrase):

Algorithm for Recognizing Clausal Complements (RCP)

(1) *If obj is a VP infinitive and the main verb subcategorizes the infinitive return T. Else assume the infinitive is a purpose infinitive and remember that fact.*

(2) *If obj is a VP ing and the main verb subcategorizes a VP ing, return T.*

(3) *If the obj is a that-clause, or a s-c (empty sbar) clause and the main verb subcategorizes a that-clause or the root-form of the verb, return T. Else return conditional T, and be ready for possible spurious argument.*

(4) *If the clause type in obj is a WHADVP and the main verb subcategorizes the conjunction return T.*

(5) *If the clause type is WHNP and the main verb subcategorizes a WHNP, return T.*

(6) *Else return NIL.*

One may wonder why in (3) the decision is not based only on the verb subcategorization, which will produce a more precise output. The reason is that the grammatical relations in the verb predicates are more complete than the verb subcategorizations in our database. For instance, the verb “add” in our database of verb subcategorizations did not include a *that-clause*. Thus, in the sentence “He added that several engineers described the invention” the algorithm is going to miss one of the arguments of “add,” and even its meaning. However, if the algorithm returns a conditional T, every thing works fine for this example, because *add2* in WordNet is a subpredicate of the predicate *state-something* whose *theme* is realized by a CP. In an ungrammatical sentence such as “He ate that several engineers described the invention,” none of the predicates for “eat” has a role realized by a CP, thus the complement clause built wrongly by the MCR, will be recognized as a spurious argument and knocked out from the structure. That clause will be ran again by the SI with the spurious argument deleted from it.

Algorithm for Choosing between Subjects (ACS)

When the MCR builds more than one subject for a clause, the algorithm (ACS) chooses between subjects based on the verb subcategorization. The general cases are given in Fig-

ure 3. Remember that potential subjects are entered in the following order: first the subject and then the *obj* of the main clause. Thus, consider the sentence “She bought a book of history to learn the truth,” discussed in section 2. The potential subjects of “learn” are “she” and “book.” The algorithm *Recognize Clausal Complements* would recognize “to learn” as a purpose infinitive in (1). The ACS algorithm in Figure 3 (step 1) selects “she” as the subject. For the sentence, “The houses were bought to be sold” the potential subjects of “sold” are “unknown-agent” and “the houses,” and the ACS algorithm will select “houses” as the *subj* (main clause is passive). For “He was told to be fair” the subjects entered for “be” are “unknown-agent” and “he,” the VP-inf “to be” is recognized as an argument not as a purpose infinitive by the *Recognize Clausal Complement* algorithm, and the ACS algorithm in Figure 3 (step2) will select the second entry in the subject slot, namely “he,” as the subject of “be.”

In step3, the ACS algorithm deals with sentences such as “Huge ice blocks prevented him from going farther.” Because “prevent” subcategorizes “from,” “him” (second entry in the *subj* slot) is selected. Again, the verb predicates are more complete than our database of verb subcategorizations. As a result, the verb predicate of the main clause (already interpreted) is accessed to find out if it subcategorizes the preposition. For instance, we failed to assign the correct subject to “giving” in “Their codes of ethics may proscribe some people from giving secret briefings” because our subcategorization for “proscribe” did not include the preposition “from.” But, “proscribe” is mapped to the verb predicate *forbid*, and the *theme* of *forbid* (thing or event being forbidden) is realized by “from” followed by a VP ing clause. The algorithm now checks if the verb predicate of the main verb subcategorizes the preposition. Thus, it is using the already built semantic interpretation of the main clause to interpret the embedded clause. Step 4 (not listed in Figure 3) defaults to the first entry in the subject slot.

Analyzing the subject of the CP in those cases in which the MCR builds only one subject

Not all infinitival CPs take their subject from the subject or the object of their main clause. For instance, “arrange” and “plan” may subcategorize infinitives whose subjects are introduced by a PP [for NP]; e.g. “Alexius arranged for Maria to stay on the palace.” Charniak’s parser does quite well in these types of verbs, recognizing perfectly the object of the PP in the main as the subject of the CP. However, Charniak’s parser does not handle well other types of verbs in which the subject of the CP is a PP headed by “with,” “to,” or other prepositions. For instance, “She pleaded with him to read the books,” “They appealed to the governor to save the trees,” or “He yelled at her to stop.” The MCR builds as subject of the CP only the subject of main, which is wrong. We have used a rule to handle these verbs, which is based on the semantic interpretation of the main clause. The rule does not make any reference to any preposition but to the roles built for the main clause. The rule is more elegant in Lisp than in English, but this is its English formulation:

- (1) If the clause type is an infinitive, and it has been recognized as a purpose infinitive by the algorithm that recognizes clausal complements (RCP) then,
 - 1.1 If the parent clause is passive, select the second entry in the subject slot.
 - 1.2 Else return the first entry in the subject slot.
- (2) If the clause type is an infinitive and it has been recognized as a complement by the RCP algorithm then,
 - 2.1 If the verb of the parent clause subcategorizes the subject (e.g. “promise”) return the first entry in the subject slot.
 - 2.2 If the verb of the parent clause subcategorizes the obj, return the second entry in the subject slot.
 - 2.3 Else return the first entry - this happens rarely when missing subcategorizations.
- (3) If the clause type is [Prep VP + ing] (e.g., “from entering,” “for forcing”), then
 - 3.1 If the main verb subcategorizes the preposition or the preposition is found in the verb predicate of the parent clause return the second entry in the subject slot.
 - 3.2 Else return the first entry in the subject slot

Figure 3: Algorithm for Choosing between Subjects (ACS)

If OBJi is infinitive or np-dominated-infinitive and the verb predicate of the main clause has communicate as its super-predicate and the main clause has a recipient role, return the recipient of the main clause as the subject of the CP.

Besides this rule, there are other rules to recognize “it,” when it stands for a clausal substitute instead of a pronoun. The first rule handles sentences such as “It was easy to copy the books.” The MCR will take “it” as subject of both “was” and “copy,” which is wrong. A rule will assign *unknown-agent* as subject of “copy” and “to copy the books” as subject of “is.” Charniak’s parser handles well the easy-construction, e.g., “It is easy for Mary to read” by recognizing “Mary” as the subject of “read.” The SI will recognize the clause “Mary to read” as the subject of “is.” Other rules handle sentences such as “It surprised her that he ate” and passive sentences in which “it” also stands as a clause substitute; e.g. “It has been said that she saved the country,” in which the subject of “said” is *unknown-agent*.

Testing the Algorithms

Unfortunately, the testing cannot be automated because of the different data structures in Ontonotes (Weischedel et al. 2011) and in the MCR. Every clause has to be manually checked for correctness by looking into the Ontonotes files. We selected randomly a letter in Ontonotes release 3 (Ontonotes release 4 had not been published yet) with at least 50 verbs, and map all senses for those verbs to the verb predicates. The letter “m” was selected, which in the release 3 of Ontonotes contains 76 verbs. We have mapped all the Ontonotes verb senses except 5 verbs (monesemous and none occurring in the WSJ annotated corpus) to verb predicates. As our definitions were ending Ontonotes release 4 was published. Then, all our testing has taken place against the annotated data in Ontonotes 4. For each verb, our program produces the WordNet (WN) senses that the annotators have matched to their verb groups. If a verb has more senses in Ontonotes than in any version of WN, our program produces the Ontonotes verb group. For some verbs, our senses are more fine grained than those in Ontonotes. We kept our

senses and judge the tested sentences correctly if our verb senses are all contained in the Ontonotes verb group selected for that verb.

We have tested each verb on at least 30 sentences. “Maximize” has been tested with only the 5 sentences in the WSJ files because we had already tested “minimize” with 31 sentences, and they have the same semantic and grammatical distribution. As of now, there are many verbs in the WSJ Ontonotes files which are not annotated with verb senses. We have selected only those sentences for which the Ontonotes annotators provide verb senses. If there were less than 30 sentences in the WSJ file, we randomly selected the remainder sentences from Wikipedia. Also, if only one verb sense was represented in the WSJ file, we selected additional Wikipedia sentences with several verb senses for the target verb. For “make” (18 senses), “move” (9 senses), and “mark” (6 senses), we selected 40 or more sentences from the WSJ files, and similar number from Wikipedia. In our homepage, one can find a directory containing all the files for each verb, and their evaluation. There is a total of 74 files because “make,” “move,” and “mark” have two files each. A total of 2505 sentences containing the target verbs have been graded. We parsed all the sentences including the ones in the Ontonotes with Charniak parser. Each verb file contains the sentences, their parses, the MCR output for the entire parse, and the verb predicate, semantic roles and adjuncts for the verb being tested, and its evaluation. Note that all clauses need to be interpreted in order to compute the roles, verb predicates and adjuncts. That is the only way on which the arguments for the target verb can be computed.

For adjuncts, we graded locative, temporal and event types, which are the most frequently occurring in the corpus tested. In the case of adjuncts, there are two problems: recognizing that they are adjuncts, and naming them as temporal, locative, beneficiary, etc. The first problem is very well defined except in the case of those roles that Pritchett (Pritchett 1992) call quasi-arguments, e.g. *instrument*. The second problem is harder because there are many adjuncts which have not been properly categorized. Grouping many of them under the label *manner* is not satisfactory. In nam-

| - | CORRECT | MISSED | WRONG |
|----------|---------|--------|-------|
| pred | 1742 | 2 | 64 |
| roles | 3243 | 92 | 68 |
| adjuncts | 247 | 20 | 16 |
| subject | 1717 | 0 | 91 |

Table 1: Values for predicates, roles, adjuncts and subject for verbs with 3 or less senses

ing adjuncts, our system provides multiple labels. Thus, for the PP “for the population” in the sentence “An economic bonanza continued for the population” the adjunct is named as *beneficiary* or *purpose*. We have graded the adjunct as correct if the first name listed is the correct one.

The SI correctly identifies *purpose infinitives* in most cases but they have not been graded. The grammatical relation *subject* has been graded and is listed on the tables. All grammatical subjects have been graded. If a subject is the wrong subject, the semantic role for that grammatical relation is not graded as wrong or correct. If a PP is weakly attached by the algorithms to the verb (it is considered an adjunct not an argument), but the parser attaches it to a NP, the attachment of the parser is preferred.

We could have selected sentences in which the heads of the constituents are not pronouns, or numbers, since we have used a very simple anaphora resolution algorithm. However, we opted against that when “it” starts a sentence. We initialized pronouns to some ontological categories, and let the SI choose. For instance, these are the ontological categories for “it:” (*it is-a (animal) (organization) (location) (physical-thing) (abstraction) (thing)*). These are the ontological categories for “one:” (*one is-a (human) (social-group) (animal) (physical-thing) (abstraction) (thing)*). “It” is of especial interest when it starts a sentence, because we want to see if the algorithms recognize “it” as a clausal substitute. Proper nouns present similar problems, not only because there are few proper nouns in WN, but some of them do not have all its ontological categories. For instance, “Hittite” appears only as a language. If the proper noun is not in WN, the following ontological categories are assumed: *human, social-group, location, written-communication*. The ontological categories assumed for common nouns (NN/NNS) not in WN are: *human, social-group, animal, physical-thing, abstraction, thing*. Because our semantic roles distinguish between “agent” (a human, a social group or animal), and *inanimate-cause* (a causal agent other than a human-agent or an animal), some arguments are labeled with those two roles, which correspond to the argument *arg0* in Ontonotes. Thus, in the sentence “The cranes destroyed the fields”, “the cranes” is labeled as *inanimate-cause* or *agent* depending on the meaning of “crane,” but in “The cranes ate the insects,” “the cranes” is labeled only as *agent*, selecting only the bird sense of “crane.”

Table 1 provide the results for all those verbs tested containing 3 or less senses. The polysemy breakdown in this group is as follows: there are 12 monosemous verbs, 22 verbs with two senses, and 21 verbs with 3 senses. The entry “pred” stands for verb predicate, or verb sense. The

| - | P | R | F1 |
|----------|-------|--------|-------|
| pred | 96.46 | 99.89 | 98.14 |
| roles | 97.95 | 97.24 | 97.59 |
| adjuncts | 93.92 | 92.51 | 93.21 |
| subject | 94.97 | 100.00 | 97.42 |

Table 2: Precision, Recall and F1 values for Table 1

| - | CORRECT | MISSED | WRONG |
|----------|---------|--------|-------|
| pred | 645 | 0 | 52 |
| roles | 1227 | 38 | 37 |
| adjuncts | 162 | 7 | 9 |
| subject | 651 | 0 | 46 |

Table 3: Values for predicates, roles, adjuncts and subject for verbs with 4 or more senses

verb sense has been identified correctly in 1742 verb occurrences out of 1808. The entries for “adjuncts” and “roles” have similar readings. The subject of the clause has been identified correctly in 1717 out of 1808 sentences for these verbs. Table 2 provides the precision, recall and F1 values for table 1.

Table 3 and 4 lists similar information for those verbs tested with 4 or more senses. In this group, the breakdown of polysemy is: 9 verbs with 4 senses, one verb with 5 senses, 3 with 6 senses, one with 8 senses, one with 9 senses, and one with 18 senses. If one compares table 2 and table 4, the entry for the verb predicates (verb senses) has the higher difference because of the greatest polysemy of verbs on table 3. But, there is not a major difference between the other entries on the table. Roles can be determined with great degree of accuracy for all verbs. Verb predicates is a harder problem. We have indicated as correct only the first verb sense, or predicate, selected by the system, but there are cases in which the second choice selected is the correct one. Note that nouns are not disambiguated because we want the system to come up with a set of preferences for noun senses, making the task of a noun sense disambiguation algorithm easier if run on the output of our system. Consider the sentence: “He is mounting the steps of a stucco building in a nearby village,” in which the verb predicate selected *mount-organize* (its theme = step1 step3 step10 - all actions) is tied to *mount-attach* which selects step4 (support, device, instrumentality) as the noun sense for its *theme*.

Related Work

FRAMENET (Fillmore et al. 2003) is a corpus annotated with semantic roles, but the semantic roles are not linked to

| - | P | R | F1 |
|----------|-------|--------|-------|
| pred | 92.54 | 100.00 | 96.13 |
| roles | 97.07 | 97.00 | 97.03 |
| adjuncts | 94.74 | 95.86 | 95.29 |
| subject | 93.40 | 100.00 | 96.59 |

Table 4: Precision, Recall and F1 values for Table 3

selectional preferences. PROPBANK (Kingsbury, Palmer, and Marcus 2002) annotates verbs with arguments. More recently Ontonotes (Weischedel et al. 2011) offers an analysis of verb senses, its matching to WordNet (Fellbaum 1998) verb senses, and an annotation of actual uses of the verbs. Ontonotes is a valuable resource not only because of the annotation but by providing a classification of verb senses into groups. A critique that could be leveled is that some of the Ontonotes group senses for verbs with high polysemy are too coarse, coalescing metaphoric and literal senses of the verbs, which are clearly distinct. Campbell's (Campbell 2004) research, the work on argument structure (Pritchett 1992) (Grimshaw 1990) and minimal commitment models (Marcus 1987; Gorrell 1991; Weinberg 1993) bear strong relation to our work.

Conclusions

In conclusion, we have presented algorithms that determine verb predicates, semantic roles and adjuncts from constituent-based parsers. The main steps in the algorithms are the construction of a clause structure for every non-auxiliary verb on the parse trees, and the refinement of the clause structure by using verb subcategorization. The final computation of verb meaning and verb arguments is done by using selectional preferences in the verb predicates, which are linked to the Wordnet noun ontology. The algorithms have been tested on 2505 sentences from Ontonotes and Wikipedia. The integration of constituent-based parser in the overall system architecture makes it possible to determine verb meaning and semantic roles for unrestricted texts. As constituent-based parsers improve, so will the overall performance of our system. The grammatical and semantic knowledge in the system can be easily refined and extended as it is run on different texts. Next, we plan to map entire articles of Wikipedia into verb predicates and its semantic roles.

Acknowledgement

This work was supported in part by the NASA Engineering and Safety Center under Grant/Cooperative Agreement NNX08AJ98A. I am very grateful to Carlos Segami for helping with implementation issues, and to Andy Schwartz and Michael Gabilondo for extracting the test sentences from Ontonotes and Wikipedia, respectively.

References

Campbell, R. 2004. Using linguistic principles to recover empty categories. In *Proc. of the 2004th Annual Meeting of the ACL*, 645–652.

Charniak, E. 2000. A maximum-entropy-inspired parser. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 132–139.

Chomsky, N. 1981. *Lectures on Government and Binding*. Dordrecht, The Netherlands: Foris.

Fellbaum, C. 1998. A semantic network of english verbs. In Fellbaum, C., ed., *WordNet: An electronic Lexical Database and some of its applications*. Cambridge, Mass: MIT Press, 1998. 69–104.

Fillmore, C. J.; Johnson, R., C.; and Petruck, M. 2003. Background to framenet. *International Journal of Lexicography* 16:235–250.

Gildea, D., and Jurafsky, D. 2002. Automatic labeling of semantic roles. *Computational Linguistics* 28(3):245–288.

Gomez, F. 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*, 87–94.

Gomez, F. 2004. Building verb predicates: A computational view. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics, ACL-04*, 351–358.

Gorrell, P. 1991. Subcategorization and sentence processing. In Berwick, R.; Abney, S.; and Tenny, C., eds., *Principle-Based Parsing: Computation and Psycholinguistics*. Dordrecht, The Netherlands: Kluwer Academic.

Grimshaw, J. 1990. *Argument Structure*. Cambridge, Mass.: MIT Press.

Kingsbury, P.; Palmer, M.; and Marcus, M. 2002. Adding semantic annotation to the penn treebank. In *Proc. of the Human Language Technology Conference*, –.

Marcus, M. 1987. Deterministic parsing and description theory. In Whitelock, P.; Wood, M.; Somers, H.; Johnson, R.; and Bennett, P., eds., *Linguistic Theory and Computer Applications*. Academic Press.

Miller, G. 1998. Nouns in wordnet. In Fellbaum, C., ed., *WordNet: An electronic Lexical Database and some of its applications*. Cambridge, Mass: MIT Press, 1998. 23–46.

Millward, C., and Gomez, F. 2010. A minimal clausal reconstruction algorithm from constituent-based parse trees. Technical report, University of Central Florida, EECS, Orlando, FL 32816. CS-TR-10-03.

Pritchett, B. L. 1992. *Grammatical Competence and Parsing Performance*. Chicago, Illinois: The University of Chicago Press.

Vickrey, D., and Koller, D. 2008. Sentence simplification for semantic role labeling. In *Proceedings of ACL-08*, 344–352.

Weinberg, A. 1993. Parameters in the theory of sentence processing: Minimal commitment theory goes east. *Journal of Psycholinguistic Research* 22(3):339–364.

Weischedel, R.; Palmer, M.; Marcus, M.; Hovy, E.; Pradhan, S.; Ramshaw, L.; Xue, N.; Taylor, A.; Kaufman, J.; Franchini, M.; and et alia. 2011. *Ontonotes release 4.0*. Philadelphia, Penn: Linguistic Data Consortium.

Winograd, T. 1983. *Language as a Cognitive Process: Volume 1, Syntax*. Reading, Mass.: Addison-Wesley Publishing Company.