Proceedings of HLT-NAACL 2003
Short Papers , pp. 100-102
Edmonton, May-June 2003

Building lexical semantic representations for Natural Language instructions

Elena Terenzi
Computer Science
Politecnico di Milano
Milano, Italy
elenat@libero.it

Barbara Di Eugenio
Computer Science
University of Illinois
Chicago, IL, USA
bdieugen@cs.uic.edu

Abstract

We report on our work to automatically build a corpus of instructional text annotated with lexical semantics information. We have coupled the parser LCFLEX with a lexicon and ontology derived from two lexical resources, VerbNet for verbs and CoreLex for nouns. We discuss how we built our lexicon and ontology, and the parsing results we obtained.

1 Introduction

This paper discusses the lexicon and ontology we built and coupled with the parser LCFLEX (Rosé and Lavie, 2000), in order to automatically build a corpus of instructional text annotated with lexical semantics information. The lexicon and ontology are derived from two lexical resources: VerbNet (Kipper et al., 2000a) for verbs and CoreLex (Buitelaar, 1998) for nouns. We also report the excellent parsing results we obtained.

Our ultimate goal is to develop a (semi)automatic method to derive domain knowledge from instructional text, in the form of linguistically motivated action schemes. To develop this acquisition engine, our approach calls for an instructional corpus where verbs are annotated with their semantic representation, and where relations such as *precondition* and *effect* between the actions denoted by those verbs are marked. Whereas the action relation annotation will be manual, the semantic annotation can be done automatically by a parser.

We are interested in decompositional theories of lexical semantics such as (Levin and Rappaport Hovav, 1992) to account for examples such as the following:

(1a) *Wipe the fingerprints from the counter.*

(1b) *Wipe the counter.*

(2a) *Remove the groceries from the bag.*

(2b) *Remove the bag.*

As the effect of the two actions (1a) and (2a), it is inferred that the specified location (*counter* in (1a), *bag* in (2a)) has been "emptied" of the object (*fingerprints* in (1a), *groceries* in (2a)). Thus, a system could map both verbs *wipe* and *remove* onto the same action scheme. However, the apparently equivalent transformations from (1a) to (1b) and from (2a) to (2b) show otherwise. (1b) describes the same action as (1a), however (2b) cannot have the same meaning as (2a). (Levin and Rappaport Hovav, 1992) defines classes of verbs according to the ability or inability of a verb to occur in pairs of syntactic frames that preserve meaning. The *location-as-object* variant is possible only with (some) *manner/means* verbs such as *wipe*, and not with *result* verbs such as *remove*.

We chose to base our lexicon and ontology on VerbNet (Kipper et al., 2000a), that operationalizes Levin's work and accounts for 960 distinct verbs classified into 72 main classes. Moreover, given VerbNet strong syntactic components, it can be easily coupled with a parser and used to automatically generate a semantically annotated corpus.

Of course, when building a representation for a sentence, we need semantics not only for verbs, but also for nouns. Whereas many NL applications use WordNet (Fellbaum, 1998), we were in need of a richer lexicon. We found CoreLex (Buitelaar, 1998) appropriate for our needs. CoreLex is based on a different theory than Levin's (that of the generative lexicon (Pustejovsky, 1991)), but does provide a compatible decompositional meaning representation for nouns.

The contribution of our work is to demonstrate that a meaning representation based on decompositional lexical semantics can be derived efficiently and effectively. We believe there is no other work that attaches a semantics of this type to a parser for a large coverage corpus. VerbNet has been coupled with the TAG formalism (Kipper et al., 2000b), but no parsing results are available. More-

```
(   :morphology   position
    :syntax       (*or*
                  ((cat n) (root position) (agr 3s) (semtag (*or* lap1 lap2)))
                  ((cat vlex) (root position) (vform bare)
                     (subcat (*or* np np-advp np-pp))(semtag put)))
    :semantics    (put (<put-9.1> (subj agent) (obj patient) (modifier destination) (pred destination)))
                  (lap1 (<lap1>))
                  (lap2 (<lap2>)))
```

Figure 1: The entry for *position* in the LCFLEX lexicon

CLASS: put-9.1
PARENT: -
MEMBERS: arrange immerse lodge mount place position put set situate sling
THEMATIC ROLES: Agt Pat Dest
SELECTIONAL RESTRICTIONS: Agt[+animate] Pat[+concrete] Dest[+location -region]
FRAMES:

Transitive with Locative PP	Agt V Pat Prep[+loc] Dest	cause(Agt, E0) ∧ motion(during(E0), Pat) ∧ ¬Located-in(start(E0), Pat, Dest) ∧ Located-in(end(E0), Pat, Dest)
Transitive with Locative Adverb	Agt V Pat Dest[+adv-loc]	cause(Agt, E0) ∧ motion(during(E0), Pat) ∧ ¬Located-in(start(E0), Pat, Dest) ∧ Located-in(end(E0), Pat, Dest)

Figure 2: The class put-9.1 from VerbNet

over, we also show that two lexical resources that focus on verbs and nouns can be successfully integrated.

2 Lexicon and ontology

We chose LCFLex (Rosé and Lavie, 2000), a robust left-corner parser, because it can return portions of analysis when faced with ungrammaticalities or unknown words or structures (the latter is likely in a large corpus). We modified and augmented LCFLEX's existing lexicon and built an ontology.

To illustrate our work, we will refer to the lexical entry for *position*, that can be both a noun (*n*) or a verb (*vlex*) – the format is provided by LCFLEX, but the `:semantics` field was originally empty (see Figure 1). For the verb, different subcategorization frames are listed under *subcat*: the verb can have as argument just an np, or an np and a pp, or an np and an adverbial phrase. Each part of speech (POS) category is associated to a *semtag*, an index that links the POS entry to the corresponding semantic representation. <put-9.1>, <lap1> and <lap2> are entries in our ontology. Before discussing the ontology, we need to discuss the VerbNet and CoreLex formalisms.

Figure 2 shows a simplified version of the VerbNet class to which the verb *position* belongs. All verbs that can undergo the same syntactic alternations belong to the same class. Each frame is labeled with its name, and consists of the syntactic frame itself (e.g., Agt V Pat Prep Dest), and its semantic interpretation. Agt stands for Agent, V for Verb, Pat for Patient, Dest for Destination. A class includes a list of parent classes, empty in this case (verb classes are arranged in a hierarchy), its thematic roles and selectional restrictions on these. Then, it specifies all the frames associated with that class, and provides a meaning representation for each frame. In this case, the two frames are both transitive. In the first the destination is a prepositional phrase, whereas in the second the destination is an adverb.

The semantics portion of a lexical entry links the syntactic roles built by the parser to the thematic roles in the verb class. In Figure 1, the following mappings are specified under `put`: subject to agent, object to patient, modifier to destination for the first frame (the parser always maps prepositional phrases to *modifier* roles), and pred to destination for the second frame (the parser usually maps adverbs to the *pred* role).

As regards nouns, CoreLex defines semantic types such as *artifact* or *information*. Nouns are characterized by bundles of semantic types. Nouns that share the same bundle are grouped in the same Systematic Polysemous Class (SPC). The resulting 126 SPCs cover about 40,000 nouns.

VerbNet classes and CoreLex SPCs are realized as entities in our ontology. Figure 3 shows the entries for `put-9.1` and the SPCs `lap2` (we omit `lap1` for lack of space). We do not have room for many details, however note that the `:spec` field is the basis for building the semantic representation while parsing. The subfields of `:spec` are structured as `(name type-check arg)`. `arg` can be either a variable or a complex argument built with one or more functions. `type-check` is a the type constraint `arg` must satisfy to be included in the final representation. For further details, see (Terenzi, 2002).

```
(:type     <put-9.1>
 :isa      nil
 :vars     (agent patient destination)
 :spec     ((agent <animate> agent)
            (patient <concr-ent> patient)
            (dest <> (<loc-not-reg> destination))
            (event <>
                (<event>
                        (<not-located-in> destination patient)
                        (<in-motion> patient)
                        (<located-in> destination patient)
                        nil
                        event))))

(:type        <lap2>
 :isa         (<loc>)
 :instances   nil
 :vars        nil
 :spec        ((artifact +)
               (location +)
               (psych-feat +)))
```

Figure 3: Two entries in our ontology

3 Results

Our lexicon includes 109 verbs and 289 nouns, grouped under 9 classes and 47 SPCs respectively (classes and SPCs are the entries in the ontology).

We evaluated LCFLEX augmented as we have described on a test set taken from the home repair portion of a 9Mb written corpus originally collected at the University of Brighton. We collected the 480 sentences that contained at least one of the verbs in our lexicon – out of 109 verbs, those sentences cover 75. These 480 sentences include a main clause plus a number of adjunct clauses. Because we were mostly interested in those specific verbs, we simplified those sentences so that the clause that contains the verb of interest becomes the main clause, and the others are discarded.

	Correct	Partially correct	Wrong	Parser error
Only Verbs	87%	4.8%	2.2%	6%
Verbs, Nouns	96%	4%	0	0

Table 1: Parsing Results

Table 1 reports our results. A correct parse means that the full semantic representation is built with every syntactic role mapped to the correct thematic role. With partial correctness we mean that e.g. not all the syntactic roles were mapped to their correct thematic roles. Correctness was judged parse by parse by one of the two authors. We conducted two evaluations, one earlier after we had not yet included nouns, and one after the full implementation. In the first evaluation (*Only Verbs*), we preprocessed the sentences so that the nouns from the corpus would be mapped to the closest noun in our then small noun lexicon of about 40 nouns. The second evaluation (*Verbs, Nouns*) was conducted on 228 sentences out of the 480 tested in the first evaluation. The 228 sentences contain the original nouns, as we now have the full lexicon for the nouns too. The improvement in the second evaluation is due to the full noun lexicon, but the absence of parser errors to improvements in a new release of the parser.

4 Conclusions and future work

We have shown that two rich lexicons such as VerbNet and CoreLex can be successfully integrated. We have also shown that a parser which uses such a lexicon and ontology performs extremely well on instructional text. We are now poised to systematically run the parser on the full home repair portion of the corpus (about 6Mb). This is likely to require additions to the lexicon and the ontology.

Acknowledgements

This work is supported by award 0133123 from the National Science Foundation. Thanks to all who shared their resources with us.

References

Paul Buitelaar. 1998. *CoreLex: Systematic Polysemy and Underspecification*. Ph.D. thesis, Computer Science, Brandeis University, February.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical DataBase*. MIT Press, Cambridge, MA.

Karin Kipper, Hoa Trang Dang, and Martha Palmer. 2000a. Class-based construction of a verb lexicon. In *AAAI-2000, Proceedings of the Seventeenth National Conference on Artificial Intelligence*, Austin, TX.

K. Kipper, H. T. Dang, W. Schuler, and M. Palmer. 2000b. Building a class-based verb lexicon using TAGs. In *TAG+5 Fifth International Workshop on Tree Adjoining Grammars and Related Formalisms*.

B. Levin and M. Rappaport Hovav. 1992. Wiping the slate clean: a lexical semantic exploration. In B. Levin and S. Pinker, editors, *Lexical and Conceptual Semantics*. Blackwell Publishers.

James Pustejovsky. 1991. The generative lexicon. *Computational Linguistics*, 17(4):409–441.

C. P. Rosé and A. Lavie. 2000. Balancing robustness and efficiency in unification-augmented context-free parsers for large practical applications. In J.-C. Junqua and G. van Noord, editors, *Robustness in Language and Speech Technology*. Kluwer Academic Press.

Elena Terenzi. 2002. Building lexical semantics representations for action verbs. Master's thesis, University of Illinois - Chicago, Decembe