

# Putting Pieces Together: Combining FrameNet, VerbNet and WordNet for Robust Semantic Parsing

Lei Shi and Rada Mihalcea

University of North Texas,  
Computer Science Department,  
Denton, TX, 76203-1366  
leishi@unt.edu rada@cs.unt.edu

**Abstract.** This paper describes our work in integrating three different lexical resources: FrameNet, VerbNet, and WordNet, into a unified, richer knowledge-base, to the end of enabling more robust semantic parsing. The construction of each of these lexical resources has required many years of laborious human effort, and they all have their strengths and shortcomings. By linking them together, we build an improved resource in which (1) the coverage of FrameNet is extended, (2) the VerbNet lexicon is augmented with frame semantics, and (3) selectional restrictions are implemented using WordNet semantic classes. The synergistic exploitation of various lexical resources is crucial for many complex language processing applications, and we prove it once again effective in building a robust semantic parser.

## 1 Introduction

The goal of a semantic parser is to identify semantic relations between words in a text, resulting in structures that reflect various levels of semantic interpretation. Such structures can be used to improve the quality of natural language processing applications by taking into account the meaning of the text. Automatic techniques for semantic parsing have been successfully used in Information Extraction and Question Answering, and are currently evaluated in other applications such as Machine Translation and Text Summarization.

The process of semantic parsing typically implies a learning stage, where the semantic structures to be identified are acquired from an existing lexical resource, which explicitly identifies the range of possible semantic relations between words in a text. While there are several lexical resources suitable for semantic parsing, built with extensive human effort over years of work – including FrameNet [3], VerbNet [5], WordNet [7], or PropBank [4] – all previous approaches to semantic parsing have relied exclusively on only one of them, as there are no connections between these resources that would enable their exploitation in a unified way. However, each resource encodes a different kind of knowledge and has its own advantages, and thus combining them together can eventually result in a

richer knowledge-base that can enable more accurate and more robust semantic parsing.

In this paper, we describe our work in integrating into a unified knowledge-base three different lexical resources: FrameNet [3] – a corpus-based frame resource; VerbNet [5] – a verb lexicon with explicit syntactic and semantic information based on Levin’s verb classes; and WordNet [7] – a semantic network encoding a large number of concepts and semantic relations. We describe the procedures used to map the various semantic constituents identified by these lexical resources (e.g. frames, semantic roles, semantic classes), and evaluate these mappings against manually annotated data. We also shortly describe a robust rule-based semantic parser that relies on this unified knowledge-base to identify the semantic structure of any open text.

## 2 Knowledge Bases for Semantic Parsing

One major problem faced by semantic parsers is the fact that similar syntactic patterns may introduce different semantic interpretations, and similar meanings can be syntactically realized in many different ways. To deal with the large number of cases where the same syntactic relation introduces different semantic relations, we need knowledge about how to map syntax to semantics. To this end, we use two main types of knowledge – about words, and about relations between words. The first type of knowledge is drawn from WordNet – a large lexical database with rich information about words and concepts. We refer to this as *word-level knowledge*. The latter is derived from FrameNet – a resource that contains information about different situations, called *frames*, and from VerbNet – a verb lexicon based on Levin’s verb classes which also provides selectional restrictions attached to semantic roles. We call this *sentence-level knowledge*.

In this section, we briefly describe each lexical resource, and show the type of information that the semantic parser extracts from these knowledge bases.

### 2.1 FrameNet

FrameNet [3] provides the knowledge needed to identify case frames and semantic roles. It is based on the theory of frame semantics, and defines a sentence level ontology. In frame semantics, a *frame* corresponds to a scenario that involves an *interaction* and its *participants*, in which participants play some kind of roles. A frame has a name, and we use this name to identify the semantic relation that groups together the semantic roles. In FrameNet, nouns, verbs and adjectives can be used to identify frames.

Each annotated sentence in FrameNet exemplifies a possible syntactic realization for the semantic roles associated with a frame for a given target word. By extracting the syntactic features and corresponding semantic roles from all annotated sentences in the FrameNet corpus, we are able to automatically build a large set of rules that encode the possible syntactic realizations of semantic frames. In our semantic parser, we use only verbs as target words for frame

identification. Currently, FrameNet defines about 3040 verbs attached to 320 different frames.

## 2.2 VerbNet

VerbNet is a verb lexicon compatible with WordNet, with explicitly stated syntactic and semantic information based on Levin’s verb classification [6]. The fundamental assumption is that the syntactic frames of a verb as argument-taking elements are a direct reflection of the underlying semantics. VerbNet associates the semantics of a verb with its syntactic frames, and combines traditional lexical semantic information such as thematic roles and semantic predicates, with syntactic frames and selectional restrictions. It explicitly implements the close relation between syntax and semantics hypothesized by Levin.

Verb entries in the same VerbNet class share common syntactic frames, and thus they are believed to have the same syntactic behavior – an important property that can be used to extend the coverage of FrameNet. Shortly, by identifying the VerbNet verb class that corresponds to a FrameNet frame, we are able to parse sentences that include verbs not covered by FrameNet. This is done by exploiting a transitivity relation via VerbNet classes: verbs that belong to the same Levin class are likely to share the same FrameNet frame, and thus their frame semantics can be analyzed even if not explicitly defined in FrameNet.

## 2.3 WordNet

WordNet [7] is the resource used to identify shallow semantic features that can be attached to lexical units. WordNet covers the vast majority of nouns, verbs, adjectives and adverbs from the English language. The words in WordNet are organized in synonym sets, called *synsets*. Each synset represents a concept. WordNet 2.0 has a large network of 152,046 words, organized in 115,420 synsets. WordNet also includes an impressive number of semantic relations defined across concepts (more than 250,000 relations in WordNet 2.0), including hypernymy/hyponymy (ISA), meronymy/holonymy (HASA), antonymy, entailment, etc.

The information encoded in WordNet is used in several stages in the parsing process. For instance, attribute relations, adjective/adverb classifications, and others are semantic features extracted from WordNet and stored together with the words, so that they can be directly used by the semantic parser. The argument constraints encoded in VerbNet (e.g. *+animate*, *+concrete*) are mapped to WordNet semantic classes, to the end of providing selectional restrictions useful for improving the frame selection and role labeling process in the semantic parser. Moreover, the mapping between WordNet verb entries and FrameNet lexical units allows for an extension of the parser coverage, by assigning common frames to verbs that are related in meaning according to the WordNet semantic hierarchies.

### 3 Combining Resources

All these resources – FrameNet, VerbNet and WordNet – have their strengths and shortcomings. In this work, we aim to combine their strengths, and eliminate their shortcomings, by creating a unified knowledge-base that links them all together, allowing them to benefit from one another.

**FrameNet** provides a good generalization across predicates using frames and semantic roles. It also includes a fairly large corpus annotated for semantic structures, which provides empirical evidence for the syntactic realization of semantic frames. This corpus can be efficiently used to learn how to identify semantic relations starting with syntactic structures. However, FrameNet does not explicitly define selectional restrictions for semantic roles. Moreover, the construction of FrameNet required significant human effort, and thus its coverage and scalability are seriously limited.

**VerbNet** instead has better coverage, and defines syntactic-semantic relations in a more explicit way. VerbNet labels thematic roles, and provides selectional restrictions for the arguments of syntactic frames. On the down side, although the verb classes in VerbNet represent powerful generalizations for the syntactic behavior of verbs, most of the times the traditional abstract thematic roles are too generic to capture a scenario similar to that represented by a semantic frame.

Finally, perhaps one of the most useful properties of **WordNet** is its almost complete coverage of English verbs, and the rich information it encodes about semantic relations between verb senses (e.g. *ISA* relations, *entailment*, *antonymy*). However, the construction of the WordNet verb lexicon is primarily based on verb meanings, and does not encode syntactic or semantic verb behavior, such as predicate-argument structures (with the exception of a very small set of about 15 typical verb structures).

With the work reported in this paper, we aim to construct a unified framework that will exploit in a synergistic way the advantages of these three different resources, and will result in a richer resource suitable for robust semantic parsing. We augment the frame semantics with VerbNet verb classes by labeling FrameNet frames and semantic roles with VerbNet verb entries and corresponding arguments. We also extend the coverage of FrameNet verbs by exploiting both VerbNet verb classes and WordNet verb synonym and hyponym relations. Moreover, we identify explicit connections between semantic roles and semantic classes, by encoding selectional restrictions for semantic roles using the WordNet noun hierarchies.

Because of the lack of connectivity between these lexical resources, previous approaches to semantic parsing have relied exclusively on only one resource. For instance, Gildea and Jurafsky [2] proposed a statistical approach based on FrameNet data for annotation of semantic roles – which is however inherently limited to those verbs covered by FrameNet. Recent work on VerbNet semantic role labeling [10] led to an unsupervised system able to identify general thematic roles with respect to VerbNet, system that unfortunately cannot be extended to the more specific FrameNet roles, since the two resources are not connected.

Finally, there is no previous work that we are aware of that combines the analysis of semantic roles and semantic classes into one single system, as there is no resource available today that would encode explicit connections between these semantic entities.

In the following sections, we illustrate the algorithms we designed to connect these three lexical resources, and briefly describe a rule-based semantic parser that relies on this unified resource for robust semantic parsing.

## 4 Connecting VerbNet to WordNet: Defining Selectional Restrictions

Selectional restrictions – as part of the VerbNet-defined semantic knowledge – are used for both semantic role identification and syntax-semantics translation. Consider for example the sentence *I break the window* versus the sentence *The hammer breaks the window*. Although the participants in the interaction *break* have identical syntactic features in both sentences, they play however different semantic roles: *I* should be identified as playing an *agent* role, while *hammer* should play the role of an *instrument*. While this distinction cannot be made based on a difference in syntactic interpretation (since both sentences have identical parse trees), a correct semantic parse can still be achieved by realizing that the two participants *I* and *hammer* belong to different ontological categories (*I* refers to a person, and *hammer* refers to a tool), and thus they are likely to play different semantic roles.

Selectional restrictions are explicitly defined in VerbNet as constraints imposed on the arguments of syntactic frames. They are specified using generic terms such as *person*, *concrete* etc., which are attached to the constituents of a syntactic frame, to the end of providing ontological information about what can play the role of an argument. In order to generalize these selectional restrictions, and benefit from the semantic classes defined in WordNet, we map the semantic constraints identified by VerbNet to ontological categories in WordNet, which are defined as collections of entities subsumed by a given semantic concept. We say that an entity *E* belongs to an ontological category *C* if the noun *E* is a child node of *C* in the WordNet semantic hierarchy of nouns. For example, if we define the ontological category for the role *instrument* (VerbNet) as *instrumentality* (WordNet), then all hyponyms of *instrumentality* can play the role of *instrument*, while other nouns such as e.g. *boy*, which are not part of the *instrumentality* category, will be rejected.

Selectional restrictions are defined using a Disjunctive Normal Form (DNF) in the following format:

$$[\text{Onto}(\text{ID}, \text{P}), \text{Onto}(\text{ID}, \text{P}), \dots], [\text{Onto}(\text{ID}, \text{P}), \dots], \dots$$

where **Onto** is a noun and *ID* is its corresponding WordNet sense, which uniquely identifies **Onto** as a node in the semantic network. *P* can be set to *p* (positive) or *n* (negative), indicating whether a noun should belong to a given category or not. For example, the restriction:

`[person(1,n),object(1,p)],[substance(1,p)]`

indicates that the noun obeying this selectional restriction should belong to `object(sense #1)` in WordNet, but not `person(sense #1)`<sup>1</sup>, or it should belong to `substance(sense #1)`.

The process of mapping VerbNet to WordNet is thus semi-automatic. We first manually link all semantic constraints defined in VerbNet (there are 36 such constraints) to one or more nodes in the WordNet semantic hierarchy. These mappings can be one-to-one, as in e.g. `+solid` → `solid(1,p)`, or one-to-many, as in e.g. `+time` → `[time_period(1,p),time_unit(1,p)]`. Next, any concepts subsumed by the ontological categories rooted by the nodes identified in WordNet are automatically mapped to the same VerbNet semantic constraints (note that this is an implicit mapping performed during the process of identifying selectional restrictions).

## 5 Connecting FrameNet to VerbNet

This second mapping problem can be divided into two main sub-tasks: (1) Mapping VerbNet verb entries to appropriate semantic frames in FrameNet; and (2) Linking arguments of VerbNet syntactic frames with corresponding FrameNet semantic roles. We have designed algorithms to automatically handle these tasks, and we evaluate their accuracy against manually annotated data.

### 5.1 Labeling VerbNet Verb Entries with Semantic Frames

In order to link a VerbNet verb entry to a FrameNet semantic frame, we need to identify corresponding verb meanings in these two lexical resources.<sup>2</sup> VerbNet verb entries are already linked to WordNet, with a list of sense IDs being attached to each verb. In order to identify the corresponding FrameNet semantic frame for each such entry, we apply a similar annotation process to the FrameNet verb lexical units, and assign each of them with WordNet senses. To ensure maximum reliability, this annotation process was performed manually. Since WordNet sense distinctions are very fine-grained, a verb entry in FrameNet or VerbNet may be linked to multiple WordNet senses, with a set of senses being regarded as a coarse sense.

The first step of the mapping algorithm consists of dividing all VerbNet verb entries into two sets, depending on whether they have a direct match in FrameNet. The division of verbs in these two sets is performed by (1) identifying

<sup>1</sup> This exclusion has to be explicitly indicated, since `person(sense #1)` is a child node of `object(sense #1)` in WordNet.

<sup>2</sup> Note however that it is not always possible to identify a FrameNet frame for a VerbNet verb entry, since VerbNet and FrameNet cover different subsets of the English verbs. Currently, VerbNet defines 4159 verb entries, while FrameNet covers 3040 verb lexical units, with 2398 verb entries being defined in both resources.

identical word forms among VerbNet verb entries and FrameNet lexical units, and (2) ensuring that these verbs share a common meaning, by checking the intersection of their corresponding sense lists.

For the first verb set, consisting of VerbNet entries that have a direct counterpart in FrameNet, we label them with the corresponding frames by relying on their common WordNet senses, as described before. Note that multiple frames can be assigned to a single VerbNet verb entry. For example, the VerbNet sense numbers 1, 2, and 3 of *admonish* belong to the verb class *advise-37.9-1*, while in FrameNet sense numbers 1 and 2 of *admonish* are defined under the frame *Attempt\_suasion* and sense number 3 under *Judgment\_direct\_address*. Hence the VerbNet entry *admonish(1,2,3)* will be labeled with both frames *Attempt\_suasion* and *Judgment\_direct\_address*.

---

**Algorithm 1: Map FrameNet frames to VerbNet verbs**

---

```

for each LU in VerbNet
  if LU defined in FrameNet under the frame F
    frame(LU) = F;
  else
    if(synonym(LU) or hypernym(LU) defined in FrameNet)
      frame(LU) = frame(synonym(LU)) or frame(hypernym(LU));
    else
      if(some other verb in the same class defined in FrameNet)
        array fo[] = frames from verbs in the same class;
        fmax = most frequently used frame in fo[];
        frame(LU) = fmax;
      else
        label with "no frame defined";

```

---

**Fig. 1.** Algorithm for mapping VerbNet verbs to FrameNet frames. LU = Lexical Unit. F = frame

For the second set, where no corresponding lexical unit can be identified in FrameNet, we have developed two algorithms that allow us to infer the correct frame. The first algorithm uses WordNet verb synonym and hyponym relations to identify frames for those verb entries for which FrameNet defines a verb with a similar meaning. The frame that a verb belongs to depends primarily on the meaning of the verb, and hence synonym and hyponym relations can be good indicators of their correct semantic frame. According to Miller [7], synonyms represent lexicalization for a common underlying concept, and therefore verbs in the same synset are likely to belong to the same semantic frame. Similarly, verbs connected by a hyponymy relation<sup>3</sup> share semantic components with their superordinate, and usually belong to a sub-frame or the same frame as the superordinate. For a verb entry of this category, the algorithm looks for its

<sup>3</sup> The hyponymy relation can be interpreted as *to V1 is to V2 in some manner*

synonym or hypernym in FrameNet and uses the frame for that verb entry. A total of 839 VerbNet verb entries were labeled with a FrameNet frame using WordNet synonym and hyponym relations.

The remaining verb entries for which we cannot find a direct correspondence in FrameNet, nor we can find a WordNet synonym and hyponym that would enable an indirect semantic correspondence, we identify the corresponding frame using a majority voting algorithm. Based on an empirical study reported by Baker and Ruppenhofer [1] that shows that Levin’s verb classes often corresponds to FrameNet frames, we assume that verbs in the same class are likely to be in the same semantic frame. The majority voting algorithm can therefore infer the most likely frame by looking at other verbs in the same class. If other verbs have been already assigned with a frame using one of the previous algorithms, we can choose the most frequently used frame in the entire verb class as the candidate for this verb entry.

The pseudo-code of the algorithm used to assign FrameNet semantic frames to VerbNet verb entries is illustrated in Figure 1.

## 5.2 Labeling VerbNet Syntactic Frame Arguments with Semantic Roles

Once each VerbNet verb entry is mapped to a semantic frame, we can also identify a mapping between FrameNet semantic roles and VerbNet arguments. VerbNet verb classes are constructed based on syntactic frames, with verbs in the same class sharing a similar syntactic behavior. Thematic roles are provided for each argument in a syntactic frame, together with selectional restrictions. The major difference between the VerbNet thematic roles and the FrameNet semantic roles is that the thematic roles are generic and global with respect to language, while the semantic roles are local and specific only to their frame. The task we have is basically to identify for each VerbNet generic thematic role a FrameNet specific semantic role.

To this end, we use features that describe the syntactic realizations of these two types of roles as the basis for the mapping process. First, we extract such features from the syntactic description of the VerbNet arguments and the FrameNet corpus annotations. Next, we try to automatically find a correspondences between a VerbNet argument and a FrameNet semantic role by comparing such syntactic features against each other. The features we currently use include *Grammatical Function (GF)* (e.g. *subject*, *object*), *Phrase Type (PT)* (e.g. noun phrase *NP*, prepositional phrase *PP*), *Voice (active or passive)*, *Head Word* of NP and PPs, and *Selectional Restriction (SR)* (defined for each argument in VerbNet). A correspondence is identified between an argument and a semantic role if their *GF*, *PT*, and *Voice* features are equivalent. If the phrase is a prepositional phrase, their prepositions should also agree. Moreover, the head word of the semantic role needs to meet the selectional restrictions of the argument.

For those verb entries that are defined in VerbNet, but not in FrameNet, we seek to identify the closest matching frame using the algorithm described in the previous section, and then use this indirect correspondence to label FrameNet



---

**Algorithm 2: Map FrameNet semantic roles to VerbNet verb arguments**

---

```

for each LU in VerbNet
  if (LU defined in frame F1)
    map_roles(LU,F1);
    if (some other verb(s) in the class has the same frame)
      for each argument of LU
        ra[] = all roles of the frame labeling the argument;
        role = most frequently used role in ra[];
    else
      if LU has a frame
        if some verb V in the same class has the same frame
          use mapping of V for this verb;
        else
          randomly choose a verb in the frame F2;
          map_roles(LU,F2);
      else
        use original thematic roles;

map_role(LU,F)
Label arguments in LU with FrameNet semantic roles for frame F

for each argument in LU
  for each role in FrameNet
    if GF,PT,Voice agree and head word meets Selectional Restriction
      this role is used to label the argument;
    else
      if no appropriate role found
        use original VerbNet thematic role

```

---

**Fig. 2.** Algorithm for mapping VerbNet roles to FrameNet roles. LU = Lexical Unit. F = frame

roles to VerbNet arguments. Finally, for the verbs that do not belong to any frame available in FrameNet, their original thematic roles are used.

The pseudo-code of the algorithm for mapping between FrameNet and VerbNet semantic roles is shown in Figure 2.

### 5.3 Evaluation

The first mapping algorithm maps FrameNet frames to VerbNet verb entries, and returns a list of structures (*Verb*, *SenseID*, *Verb-Class*, *Frame*), where *Verb* represents the word form of the verb, *SenseID* is the WordNet sense number denoting a particular meaning of the verb, *Verb-Class* is the VerbNet class it belongs to, and *Frame* is the FrameNet semantic frame. To evaluate this first algorithm, we checked the automatically produced mappings against a manually constructed list of such structures produced for a random subset of 400 verbs,

and found that 81.25% of the VerbNet entries were assigned with a correct FrameNet frame.

The second mapping algorithm identifies explicit connections between VerbNet arguments and FrameNet semantic roles, and returns a list of structures (*Verb*, *SenseID*, *Role\_with\_Syntax\_List*). The *Role\_with\_Syntax\_List* field defines how arguments in the predicate-argument structure should be labeled with semantic roles based on their syntactic features. Basically, for each argument in the VerbNet predicate-argument structure, we identify a list of syntactic features (*GF*, *PT*, *Voice*, as described before), together with the corresponding selectional restriction and the FrameNet semantic role. We manually checked a randomly selected subset of 400 such mapped arguments, and found that 78.22% were labeled correctly with their corresponding FrameNet semantic role.

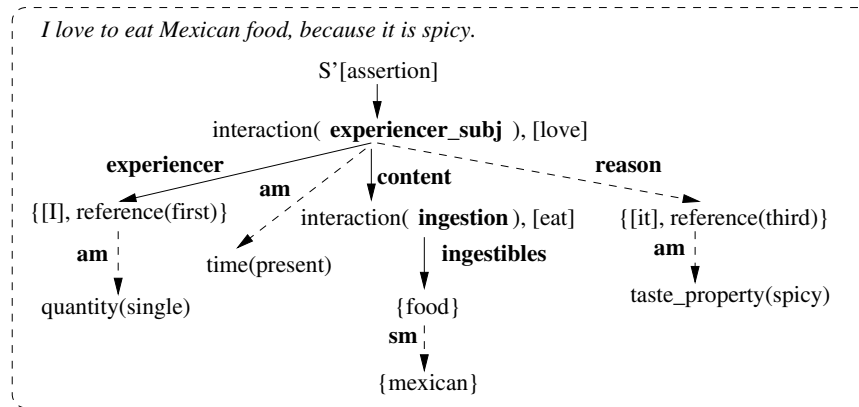
## 6 Semantic Parsing for Open Text

In this section, we briefly describe a rule-based semantic parser that relies on the unified knowledge-base that integrates FrameNet, VerbNet, and WordNet. A more detailed description of the parser can be found in [8, 9].

Similar in spirit with the syntactic parser – whose goal is to parse a valid natural language sentence into a parse tree indicating how the sentence can be syntactically decomposed into smaller syntactic constituents – the purpose of the semantic parser is to analyze the structure of sentence meaning. Sentence meaning is composed by *entities* and *interactions* between entities, where entities are assigned semantic *roles*, and entities and the interaction can be further modified by other *modifiers*. The meaning of a sentence is decomposed into smaller semantic units connected by various semantic relations by the principle of compositionality, and the parser represents the semantic structure – including semantic units as well as semantic relations, connecting them into a formal format.

For instance, the meaning of the sentence *I love to eat Mexican food, because it is spicy* can be decomposed into *I* and *to eat Mexican food* as participants in the interaction denoted by *love*; *because it is spicy* modifies the interaction *love* indicating *reason* of why this interaction takes place, and *spicy* modifies *it* indicating an attribute of *taste\_property* for the entity *Mexican food*. Finally, *eat Mexican food* can be further decomposed as *eat* representing the interaction between *Mexican food* and a hidden participant *I*, with *Mexican* being a restrictive modifier for *food*. The semantic structure is recursive, and the complete semantic tree built by the semantic parser for this sample sentence is shown in Figure 3.

Our semantic parser – which we call SPOT (Semantic Parsing for Open Text) – is a rule-based parser that integrates both word and sentence level information, extracted from WordNet, VerbNet, and FrameNet. The semantic parsing process consists of four main steps: (1) A syntactic-semantic analyzer analyzes the syntactic structure, and uses lexical semantic knowledge to identify some semantic relations between constituents. It also prepares syntactic features for semantic role assignment in the next step. (2) The corresponding FrameNet



**Fig. 3.** Semantic parse tree for a sample sentence (*am* = attributive modifier, *rm* = referential modifier, *sm* = restrictive modifier)

frame is identified, either as a direct match, or indirectly via VerbNet and/or WordNet relations, using the mapping algorithms illustrated in the previous section. Selectional restrictions are also identified and added as features. (3) The role assigner labels semantic roles for identified participants, based on their syntactic features and selectional restrictions, as produced in the first two steps. (4) For those constituents not exemplified in FrameNet, we apply default rules to decide their default meaning.

Details on each of these steps, illustrative examples, and evaluation results are provided in [8].

By relying on knowledge extracted from several lexical resources, our semantic parser has several advantages as compared to previously developed parsers e.g. [2], [10]. First, it has significantly larger coverage, being able to parse any English sentence whose target verb is defined in either FrameNet, VerbNet, or WordNet. Second, the parser is able to identify additional semantic properties, such as attribute relations, adjective/adverb classifications, etc., which are explicitly encoded in WordNet. Finally, in addition to labeling the arguments of an interaction with semantic roles, the parser also identifies selectional restrictions linked to WordNet semantic classes, which makes the labeling process more general, and opens the doors to new text analysis applications such as the use of semantic parsing for word sense disambiguation, semantic information extraction, learning of semantic classes, and others.

## 7 Conclusions

Building accurate lexical resources requires extensive human effort. Each resource is usually intended to solve a particular type of problems, and may have strengths in some aspects and shortcomings in others. In this paper, we described our work in combining three lexical resources – FrameNet, VerbNet, and

WordNet – into a unified, richer knowledge-base<sup>4</sup>. By linking all these resources together, we built an improved resource in which (1) the coverage of FrameNet is extended, (2) the VerbNet lexicon is augmented with frame semantics, and (3) selectional restrictions are implemented using WordNet semantic classes. The synergistic exploitation of these lexical resources was found effective in building a robust semantic parser.

## Acknowledgments

We thank Srinivasan Vaidyaraman for his help with the annotation and validation process. This work was partially supported by a National Science Foundation grant IIS-0336793.

## References

1. BAKER, C., AND RUPPENHOFER, J. Framenet’s frames versus Levin’s verb classes. In *Proceedings of the 28th Annual Meeting of the Berkeley Linguistics Society* (2002).
2. GILDEA, D., AND JURAFSKY, D. Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Conference of the Association for Computational Linguistics (ACL 2000)* (Hong Kong, October 2000), pp. 512–520.
3. JOHNSON, C., FILLMORE, C., PETRUCK, M., BAKER, C., ELLSWORTH, M., RUPPENHOFER, J., AND WOOD, E. *FrameNet: Theory and Practice*. 2002. <http://www.icsi.berkeley.edu/framenet>.
4. KINGSBURY, P., PALMER, M., AND MARCUS, M. Adding semantic annotation to the Penn TreeBank. In *Proceedings of the Human Language Technology Conference HLT-2002* (San Diego, California, 2002).
5. KIPPER, K., H.T.DANG, AND PALMER, M. Class-based construction of a verb lexicon. In *Proceedings of Seventeenth National Conference on Artificial Intelligence AAAI 2000* (Austin, TX, July 2000).
6. LEVIN, B. *English Verb Classes and Alternation: A Preliminary Investigation*. The University of Chicago Press, 1993.
7. MILLER, G. Wordnet: A lexical database. *Communication of the ACM* 38, 11 (1995), 39–41.
8. SHI, L., AND MIHALCEA, R. An algorithm for open text semantic parsing. In *Proceedings of the ROMAND 2004 workshop on “Robust Methods in Analysis of Natural language Data”* (Geneva, Switzerland, August 2004).
9. SHI, L., AND MIHALCEA, R. Semantic parsing using FrameNet and WordNet. In *Proceedings of the Human Language Technology Conference (HLT/NAACL 2004)* (Boston, May 2004).
10. SWIER, R., AND STEVENSON, S. Unsupervised semantic role labelling. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)* (Barcelona, Spain, 2004).

---

<sup>4</sup> Data available from <http://lit.csci.unt.edu/~spot>