Learning a Lexicon for Broad-Coverage Semantic Parsing

James F. Allen
Dept. of Computer Science, University of Rochester
james@cs.rochester.edu

Abstract

While there has been significant recent work on learning semantic parsers for specific task/ domains, the results don't transfer from one domain to another domains. We describe a project to learn a broad-coverage semantic lexicon for domain independent semantic parsing. The technique involves several bootstrapping steps starting from a semantic parser based on a modest-sized hand-built semantic lexicon. We demonstrate that the approach shows promise in building a semantic lexicon on the scale of WordNet, with more coverage and detail that currently available in widely-used resources such as VerbNet. We view having such a lexicon as a necessary prerequisite for any attempt at attaining broad-coverage semantic parsing in any domain. The approach we described applies to all word classes, but in this paper we focus here on verbs, which are the most critical phenomena facing semantic parsing.

1. Introduction and Motivation

Recently we have seen an explosion of work on learning semantic parsers (e.g., Matuszek, et al, 2012; Tellex et al, 2013; Branavan et al, 2010, Chen et al, 2011). While such work shows promise, the results are highly domain dependent and useful only for that domain. One cannot, for instance, reuse a lexical entry learned in one robotic domain in another robotic domain, let alone in a database query domain. Furthermore, the techniques being developed require domains that are simple enough so that the semantic models can be produced, either by hand or induced from the application. Language in general, however, involves much more complex concepts and connections, including discussion of involves abstract concepts, such as plans, theories, political views, and so on. It is not clear how the techniques currently being developed could be generalized to such language.

The challenge we are addressing is learning a broad-coverage, domain-independent semantic parser, i.e., a semantic parser that can be used in any domain. At present, there is a tradeoff between the depth of semantic representation produced and the coverage of the techniques. One of the critical gaps in enabling more general, deeper semantic systems is the lack of any broad-coverage deep semantic lexicon. Such a lexicon must contain at least the following information:

i. an enumeration of the set of distinct senses for the word (e.g., as in WordNet, PropBank), linked into an ontology that supports reasoning
ii. For each sense, we would have
   - *Deep argument structure*, i.e., semantic roles with selectional preferences
   - *Constructions* that map syntax to the deep argument structure (a.k.a. linking rules)
   - *Lexical entailments* that characterize the temporal consequences of the event described by the verb

The closest example to such lexical entries can be found in VerbNet (Kipper et al, 2008), a hand-built resource widely used for a range of general applications. An example entry from VerbNet is seen in Figure 1, which describes a class of verbs called murder-42.1. VerbNet clusters verbs by the constructions they take, not by sense or meaning, although many times, the set of constructions a verb takes is a good feature for clustering by semantic meaning. We see that the verbs in this class can take an AGENT, PATIENT and INSTRUMENT role, and we see the possible constructions that map syntactic structure to the deep argument structure. For instance, the first entry indicates that the simple transitive construction has the AGENT as the subject and the PATIENT as the object. In addition, it specifies lexical entailments in an informal notation, roughly stating that murder verbs involve causing a event that is a transition from being alive to not being alive. Unfortunately, VerbNet only covers a few thousand verbs. This paper reports on work to automatically build entries with much greater coverage and more detail than found in VerbNet, for all the senses in WordNet. This includes the deep argument structure and constructions for each sense, as well as axioms describing lexical entailments, expressed in a formally defined

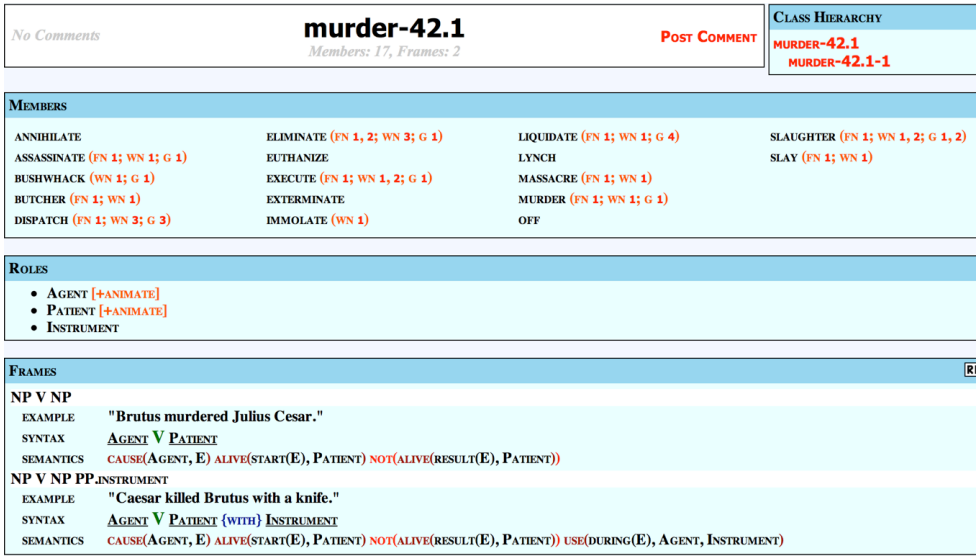

Figure 1: VerbNet Entry for murder

temporal logic (Allen, 1984; Allen & Teng, 2013).

2. Overview of the Approach

To attain broader coverage of the verbs (and their senses) for English, we look to WordNet. Though WordNet has excellent coverage, it does not contain information about argument structure, and has varying quality of ontological information (good for nouns, some information for verbs, and little for adjective and adverbs). But it does contain rich sources of information in unstructured form, i.e., each sense has a gloss that defines the word's meaning, and often provides examples of the word's usage. The technique we describe here uses an existing hand-built, but relatively limited coverage, semantic lexicon to bootstrap into a comprehensive lexicon by processing these definitions and examples. In other words, we are learning the lexicon by reading the dictionary.

Specifically, we use the TRIPS parser (Allen et al, 2008) as the starting point, which has a semantic lexicon of verbs about the same size as VerbNet. To build the comprehensive semantic lexicon, we use two bootstrapping steps. The first uses ontology mapping techniques to generate underspecified lexical entries for unknown words. This technique enables the parser to construct interpretations of sentences involving words not encoded in the core lexicon. We then use information extracted from the definitions and examples to build much more detailed and deeper lexical entries. We have run this process over the entire set of WordNet entries and provide preliminary results below evaluating the results along a number of key dimensions.

2.1. The TRIPS Parsing System

The TRIPS system is a packed-forest chart parser which builds constituents bottom-up using a best-first search strategy (Allen et al, 2008). The core grammar is a hand-built, lexicalized context-free grammar, augmented with feature structures and feature unification, and driven by a semantic lexicon and ontology. The core semantic lexicon[1] was constructed by hand and contains more than 7000 lemmas, For each word, it specifies its possible senses (i.e., its ontology type), and for each sense, its semantic roles and semantic preferences, and constructions for mapping from syntax to semantics.

The system uses variety of statistical and preprocessors to improve accuracy. These include the Stanford tools for POS tagging, named entity recognition and syntactic parsing. The parser produces and detailed logical form capturing the semantics of the sentence in a graphical notation equivalent to an unscoped, modal logic (Manshadi et al, 2012).

2.2. Level One Bootstrapping: Generating Lexical Entries Foe Unknown Words

The key idea in generating abstract lexical entries for unknown verbs builds from the same intuition the motivations underlying VerbNet -

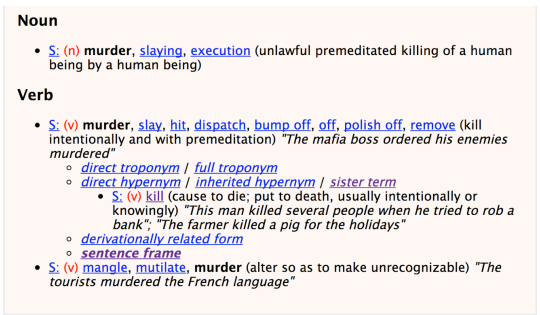

Figure 2: WordNet Entry for murder

1 you can browse the lexicon and ontology at www.cs.rochester.edu/research/trips/lexicon/browse-ont-lex.html

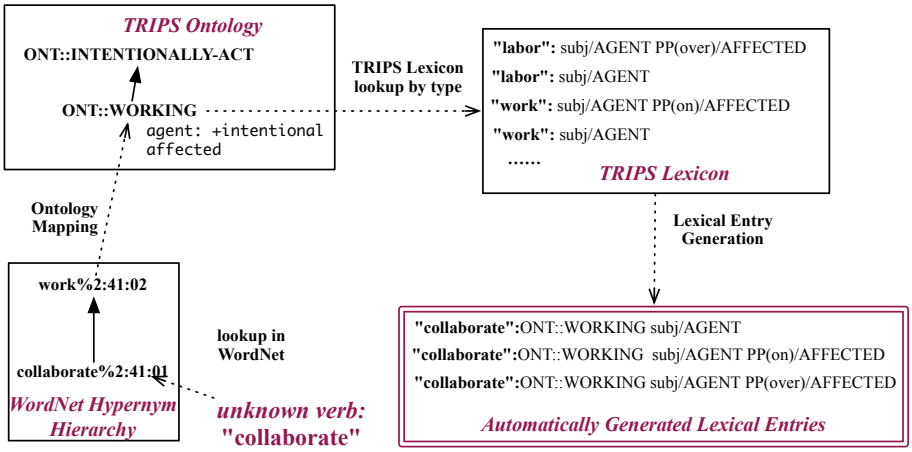

Figure 3: Example of Ontology-based Automatic Lexicon Generation

that the set of constructions a verb supports reflects its semantic meaning. While in VerbNet, the constructions are used to cluster verbs into semantic classes, we work in the opposite direction and use the semantic classes to predict the likely syntactic constructions.

To generate the lexical entries for an unknown verb we use the synset hierarchy in WordNet, plus a hand-built mapping between certain key synsets and the classes in the TRIPS ontology. The whole process operates as follows, given an unknown word *w:*

i. Look up word *w* in WordNet and obtain its possible synsets
ii. For each synset, find a mapping to the TRIPS ontology
    i. If there is a direct mapping, we are done
    ii. If not, traverse up the WordNet Hypernym hierarchy and recursively check for a mapping
iii. For each TRIPS ontology type found, gather all the words in the TRIPS lexicon that are associated with the type
iv. Take the union of all the constructions defined on the words associated with the TRIPS type
v. Generate a lexical entry for each possible combination of constructions and types

The result of this process is an over-generated set of underspecified lexical entries. Figure 3 illustrates this with a very simple example of deriving the lexical entries for the verb "collaborate": it is first looked up in WordNet, then we traverse the hypernym hierarchy until we find a mapping to the TRIPS ontology, from work%2:41:02 to ONT::WORKING. From there we find all the lexical entries associated with ONT::WORKING, and then take the union of the lexical information to produce new entries. The valid entries will be the ones that contribute to successful parses of the sentences involving the unknown words. In addition to what is shown, other lexical information is also derived in the same way, including weak selectional preferences for the argument roles.

While the result of this stage of bootstrapping produces lexical entries that identify the TRIPS type, the semantic roles and constructions, many of the lexical entries are not valid and not very deep. In particular, even considering just the correct entries, the semantic models are limited to the relatively small TRIPS ontology, and do not capture lexical entailments. Also, the selectional preferences for the semantic roles are very weak. These problems are all addressed in the second bootstrapping step.

2.3. Level Two Bootstrapping: Reading Definitions and Examples

The key idea in this stage of processing is to use the lexicon bootstrapped in level one to parse all the definitions and examples for each WordNet synset. We then use this information to build a richer ontology, better identify the semantic roles and their selectional preferences, and identify the appropriate constructions and lexical entailments. The hope is that the result is this process will be lexical entries suitable for semantic parsing, and tightly coupled with an ontology and commonsense knowledge base suitable for reasoning.

Consider an example processing a sense of the verb *keep up*, defined as *prevent from going to bed at night*. We use sense tagged glosses obtained from the Princeton Gloss Corpus to provide guidance to the parser. The TRIPS parser produces the logical form for the definition as shown in Figure 4. Each node in the graph specifies the most specific TRIPS ontology class that covers the word plus the WordNet sense. For example, the verb *prevent* is captured by a node indicating its WordNet sense prevent%2:41:00 and the TRIPS class ont::HINDERING. Note the verb *go to bed,* tagged as a multi-word verb in the Gloss corpus, has no information in the TRIPS ontology other than being an event of some sort. The semantic roles are indicated by the labelled arcs between the nodes. The nodes labelled IMPRO are the elided arguments in the definition (i.e., the missing subject and object).

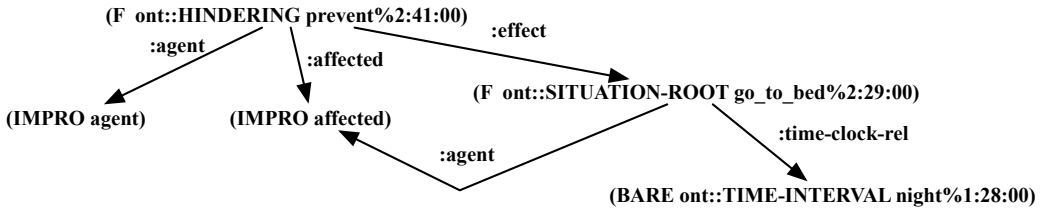

Figure 4: The parse of prevent from going to bed at night

From this definition alone we can extract several key pieces of semantic information about the verb *keep up*, namely[2]

i. Ontological: *keep_up* is a subclass of prevent %2:41:00 and ont::HINDERING events
ii. Argument Structure: *keep_up* has two semantic roles: AGENT and AFFECTED[3]
iii. Lexical Entailment: When a *keep_up* event occurs, the AGENT prevents the AFFECTED from going to bed

Definitions can be notoriously terse and complex to parse, and thus in many cases the parser can only extract key fragments of the definition. We use the TRIPS robust parsing mechanism to extract the most meaningful parse fragments when a complete parse cannot be found.

To identify the selectional preferences for the roles and the valid constructions, we parse the examples given in WordNet, plus synthetically produced sentences derived from the WordNet *sentence frame* information, plus additional examples extracted from SEMCOR, in which the words are tagged with their WordNet senses. From parsing the examples, we obtain a set of examples of the semantic roles used, plus *the constructions used to produce them.* We apply a heuristic process to combine the proposed role sets from the definitions and the glosses to arrive at a final role set for the verb. We then gather the semantic types of all the arguments from the examples, and abstract them using the derived ontology to produce the most compact set of types that cover all the examples seen. Here we present a few more details of the approach.

Determining Semantic Roles

One of the interesting observations that we discovered in this project is that the missing parts of definitions are highly predictive of the roles of the verb being defined. For instance, looking at Figure 4, we see that the verb *prevent,* used in the definition, has three roles: AGENT, AFFECTED, and EFFECT. Two of these are filled by implicit pro (**IMPRO**) forms (i.e., they were elided in the definition), and one is fully instantiated. Almost all the time it is the IMPRO roles that are promoted be the roles of *keep up*. We have found this technique to be highly reliable when we have fully accurate parsing. Because of the inevitable errors in parsing such terse language, however, we find the combining the information from the definition with additional evidence produced by parsing concrete examples gives better accuracy.

Computing Lexical Entailments

To compute lexical entailments, we use the definitions, often expanding them by recursively expanding the senses in the definition with their definitions. At some stage, the definitions of certain verb verbs become quite abstract and/or circular. To deal with this, we hand coded axiomatic definitions for a small set of aspectual verbs such as *start*, *end*, and *continue*, and causal verbs such as *cause, prevent, stop*, in a temporal logic. When a definition is expanded to the point of including one of these verbs, we can create a "temporal map" of entailments from the event. Thus, from the definition of *keep up*, we can infer that the event of going to bed does not occur over the time over which the *keep up* event occurs. A description of our first attempt to generate entailments can be found in Allen et al (2011), and the temporal logic we have developed to support compositional derivation of entailments is described in Allen & Teng (2013).

Computing Selectional Preferences

We compute selectional preferences by gathering the ontological types of elements that fill each argument position, using examples drawn from WordNet and SEMCOR. We then generalize this set by trying to find non-trivial subsuming types that cover the examples. For example, for the verb *kill*, we might find examples of the AFFECTED role of being a *person*, a *pig*, and a *plant.* We try to find a subsuming type that covers all of these classes that is more specific than the extremely abstract classes such as

2 The ontology is represented in OWL-DL (www.w3.org/TR/owl-guide), and the entailments in a logic based on Allen's (1984) Logic of Action and Time. There is no space to present these details in this paper.

3 The AFFECTED role in TRIPS includes most cases using the PATIENT role in VerbNet

REFERENTIAL-SEM (the class of all things that can be referred to). We compute this over a combined ontology using the TRIPS ontology plus the ontology that we derive from parsing all the WordNet definitions. Using both allows us to avoid the pitfalls of lack of coverage in one source or the other. As an example, in this case we would find the class LIVING-THING covers the three examples above, so this would be the derived selectional preference for this role of *kill*. Selectional preferences derived by the method have been shown to be useful in automatically identifying metaphors (Wilks et al, 2013).

3. Evaluations

This is a work in progress, so we do not yet have a comprehensive evaluation. We do have preliminary evaluations of specific aspects of the lexical entries we are producing, however. For the most part, our evaluations have been performed using set of human judges (some fellow colleagues and some recruited using Amazon Turk). Because of the complexity of such judging tasks, we generally use at least seven judges, and sometimes up to eleven. We then eliminate cases where there is not substantial human agreement, typically at least 75%. We have found that this eliminates less that 20% of the potential test cases. The remaining cases provide a gold standard.

The Event Ontology

To evaluate the derived event ontology, we randomly created a evaluation set consisting of 1) subclass pairs derived by our system, 2) hypernym pairs extracted from WordNet, and 3) random pairs of classes. We used eleven human judges to judge whether one class is a subclass of the other, and evaluated the system on the cases where at least eight judges in agreement (83% of cases). The system had 83% precision and 42% recall in this test, indicating good accuracy. The low recall score, however, indicates our techniques do not extract many of the hypernym relations present in WordNet. It suggests that we should also incorporate the hypernym relations as a ontology source when constructing the final deep semantic lexicon. More details can be found in Allen et al (2013).

Causal Relations Between Events

We used a similar technique to evaluate our ability to extract causal relationships between events classes (e.g., *kill* causes *die*). We tested on a similar blend of derived casual relations, explicitly annotated causal relations in WordNet and random other pairs. The system achieved 100% precision and 55% recall on this test. Interestingly, there was almost no overlap between the system-derived causal relations and those in WordNet, indicating that combining the two sources will produce a much richer resource. More details can be found in Allen et al (2013).

Selectional Preferences for Roles

We performed a preliminary evaluation on the correctness of the selectional preferences by comparing our derived classes with the restrictions in VerbNet. This is not an ideal evaluation as the VerbNet restrictions are quite abstract. For instance, VerbNet has one class for abstract objects, whereas the our derived ontology has a much richer classification, including plans, words, properties, beliefs, and so on. Thus, we expected that often our derived preferences would be more specific than the VerbNet restrictions. On a test set of 50 randomly selected verbs, 51% of the restrictions were exactly correct, 26% were too specific, 19% too general, and 2% were inconsistent. These results suggest promise for the approach. We are designing a more refined experiment using human judges to attempt to drill deeper.

4. Conclusion

The preliminary evaluations are promising and suggest it could be feasible to automatically build a deep semantic lexicon on the scale of WordNet, tightly integrated with an ontology also derived from the same sources. We are continuing this work in a number of directions, and designing better evaluation metrics. In addition, as many researchers find the WordNet inventory of word senses too fine grained, we are developing techniques that used the derived information to automatically cluster sets of senses in more abstract senses that cover them.

When the project is completed, we will be releasing the full semantic lexicon for use by other researchers.

As a final note, while the TRIPS system is an essential part of the bootstrapping process, it is trivial to remove all traces of TRIPS in the final resource, removing the hand-built lexical entries and the TRIPS ontology, leaving a resource entirely grounded in WordNet.

5. Acknowledgements

This paper summarizes work performed over a several years involving a large group of people including William de Beaumont, Hyuckchul Jung, Lucian Galescu, Janson Orfan, Mary Swift, and Choh Man Teng. This work was supported in part by NSF grant 0958193, ONR grant N000141110417, and the Nuance Foundation.

6. References

Allen, J. F. (1984). "Towards a General Theory of Action and Time." Artifical Intelligence 23: 123-154.

Allen, J., M. Swift and W. de Beaumont (2008). Deep Semantic Analysis for Text Processing. Symposium on Semantics in Systems for Text Processing (STEP 2008)

Allen, J., W. de Beaumont, et al. (2011). Acquiring Commonsense Knowledge for a Cognitive Agent. AAAI Symposium on Advances in Cognitive Systems, Washington, DC.

Allen, J., W. de Beaumont, et al. (2013). Automatically Deriving Event Ontologies for a CommonSense Knowledge Base. Proc. 10th International Conference on Computational Semantics (IWCS 2013), Potsdam, Germany.

Allen, J. and C. M. Teng (2013). Becoming Different: A Language-driven formalism for commonsense knowledge. CommonSense 2013: 11th Intl Symposium on Logical Formalization on Commonsense Reasoning, Cypress.

Branavan, S., L. Zettlemoyer, and R. Barzilay. (2010) Reading between the lines: Learning to map high-level instructions to commands. In ACL, pages 1268–1277.

Chen D.L., and R. Mooney.(2011) Learning to interpret natural language navigation instructions from observations. Proc. AAAI (AAAI-2011), pages 859–865

Dzikovska, M., J. F. Allen, et al. (2008). "Linking Semantic and Knowledge Representation in a Multi-Domain Dialogue System." Logic and Computation 18(3): 405-430.

Fellbaum, S. (1998, ed.) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Pr

Kipper, K, A Korhonen, N Ryant, M Palmer. (2008) "A Large-scale Classification of English Verbs." Language Resources and Evaluation Journal,42(1).

Manshadi, M. and J. Allen (2012a). A Universal Representation for Shallow and Deep Semantics. LREC Workshop on Semantic Annotation. Instanbul, Turkey.

Matuszek, C., E Herbst, L Zettlemoyer, D. Fox (2012), Learning to Parse Natural Language Commands to a Robot Control System, Proc. of the 13th International Symposium on Experimental Robotics (ISER)

Tellex, S., P Thaker, J Joseph, N Roy. (2013). Learning Perceptually Grounded Word Meanings From Unaligned Parallel Data. Machine Learning Journal

Wilks, Y., L. Galescu, et al. (2013). Automatic Metaphor Detection using Large-Scale Lexical Resources and Conventional Metaphor Extraction. Proceedings of the First Workshop on Metaphor in NLP, Atlanta, GA.