

# Learning to Map Text to Graph-based Meaning Representations via Grammar Induction

Smaranda Muresan

Laboratory for Computational Linguistics and Information Processing

Institute for Advanced Computer Studies

University of Maryland

College Park, MD 20742, USA

smara@umiacs.umd.edu

## Abstract

We argue in favor of using a graph-based representation for language meaning and propose a novel learning method to map natural language text to its graph-based meaning representation. We present a grammar formalism, which combines syntax and semantics, and has ontology constraints at the rule level. These constraints establish links between language expressions and the entities they refer to in the real world. We present a relational learning algorithm that learns these grammars from a small representative set of annotated examples, and show how this grammar induction framework and the ontology-based semantic representation allow us to directly map text to graph-based meaning representations.

## 1 Introduction

Recent work (Wong and Mooney, 2007; Zettlemoyer and Collins, 2005; He and Young, 2006) has developed learning algorithms for the problem of mapping sentences to their underlying semantic representations. These semantic representations vary from  $\lambda$ -expressions (Bos et al., 2004; Zettlemoyer and Collins, 2005; Wong and Mooney, 2007) to DB query languages and command-like languages (RoboCup Coach Language, CLang) (Ge and Mooney, 2005).

In this paper we focus on an ontology-based semantic representation which allows us to encode the meaning of a text as a direct acyclic graph. Recently, there is a growing interest on ontology-based NLP, starting from efforts in defining ontology-based semantic representations

(Nirenburg and Raskin, 2004), to using ontological resources in NLP applications, such as question answering (Basili et al., 2004; Beale et al., 2004), and building annotated corpora, such as the OntoNotes project (Hovy et al., 2006).

There are three novel properties to ontology-based semantics that we propose in this paper:

- There is a *direct link* between the ontology and the grammar through constraints at the grammar rule level. These ontology constraints enable access to meaning during language processing (parsing and generation).
- Our ontology-based semantic representation is expressive enough to capture various phenomena of natural language, yet restrictive enough to facilitate *grammar learning*. The representation encodes both ontological meaning (concepts and relations among concepts) and extra-ontological meaning, such as voice, tense, aspect, modality.
- Our representation and grammar learning framework allow a direct mapping of text to its meaning, encoded as a direct acyclic graph (DAG). We consider that “understanding” a text is the ability to correctly answer, at the conceptual level, all the questions asked w.r.t to that text, and thus *Meaning = Text + all Questions/Answers* w.r.t that *Text*. Under this assumption, obtaining the meaning of a text is reduced to a question answering process, which in our framework is a *DAG matching problem*.

First, we review our grammar formalism introduced in (Muresan, 2006; Muresan and Rambow, 2007), called *Lexicalized Well-Founded Grammars*. Second, we present a relational learning algorithm for inducing these grammars from a representative sample of strings annotated with their semantics, along with minimal assumptions about

© 2008. Licensed under the *Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported* license (<http://creativecommons.org/licenses/by-nc-sa/3.0/>). Some rights reserved.

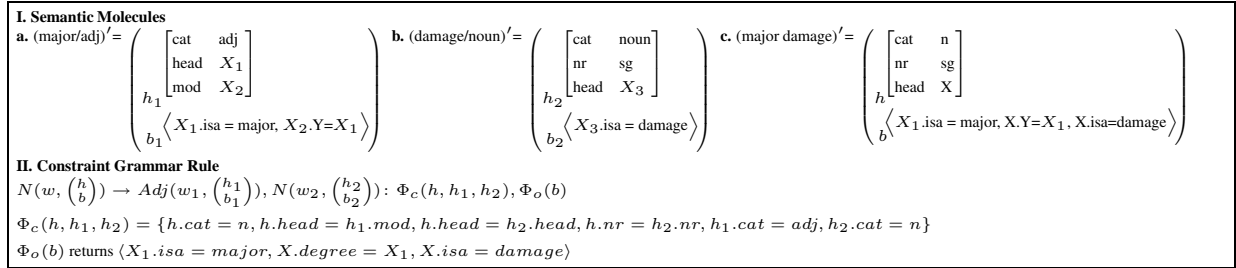


Figure 1: Examples of three semantic molecules (I), and a constraint grammar rule together with the semantic composition and ontology-based interpretation constraints,  $\Phi_c$  and  $\Phi_o$  (II)

syntax. Then, we describe the levels of representation we use to go from utterances to their graph-based meaning representations, and show how our representation is suitable to define the meaning of an utterance/text through answers to questions. As a proof of concept we discuss how our framework can be used to acquire terminological knowledge from natural language definitions and to query this knowledge using wh-questions.

## 2 Grammar Formalism

Lexicalized Well-Founded Grammars (LWFGs) introduced in (Muresan, 2006; Muresan and Rambow, 2007) are a type of Definite Clause Grammars (Pereira and Warren, 1980) where: (1) the context-free backbone is extended by introducing a partial ordering relation among nonterminals (the basis for “well-founded”); (2) each string is associated with a syntactic-semantic representation called a *semantic molecule*; and (3) grammar rules have two types of constraints: one for semantic composition and one for ontology-based semantic interpretation. The last two properties allow us to have a syntactic-semantic grammar. The ontology constraints provide access to meaning during language learning, parsing and generation. The first property allows us to learn these grammars from a small set of annotated examples.

The *semantic molecule* is a syntactic-semantic representation of natural language strings  $w' = \binom{h}{b}$ , where  $h$  (*head*) encodes the information required for semantic composition, and  $b$  (*body*) is the actual semantic representation of the string. Figure 1 gives examples of semantic molecules for an adjective, a noun and a noun phrase, as presented in (Muresan and Rambow, 2007).

The head  $h$  of the semantic molecule is a flat feature structure (i.e., feature values are atomic), having at least two attributes that encode the syntactic category of the associated string, `cat`, and

the head of the string, `head`. In addition, attributes for agreement and other grammatical features can be present (e.g., `nr`, `pers` for number and person). The set of attributes is finite and known a-priori for each syntactic category. Being a one-level feature structure, no recursive or embedded structures are allowed (unlike other grammar formalisms such as HPSG, LFG), which makes this representation appealing for a learning framework. Recursion in the grammar is obtained through the recursive grammar rules and the composition constraint.

The body,  $b$ , of a semantic molecule is a flat representation, called *OntoSeR* (Ontology-based Semantic Representation). No embedding of predicates is allowed, as in Minimal Recursion Semantics (MRS) (Copestake et al., 1999). Unlike MRS, *OntoSeR* is a logical form built as a conjunction of *atomic predicates*  $\langle concept \rangle. \langle attr \rangle = \langle concept \rangle$ , where variables are either concept or slot (`attr`) identifiers in an ontology. For example, the adjective *major* is represented as  $\langle X_1.isa = major, X_2.Y = X_1 \rangle$ , which says that the meaning of an adjective is a concept  $X_1$  ( $X_1.isa = major$ ) that is the value of a property of another concept  $X_2$  ( $X_2.Y = X_1$ ) in the ontology.

A LWFG specifies one or more semantic molecules for each string that can be parsed by the grammar. The lexicon of a LWFG consists of words paired with their semantic molecules shown in Figure 1(Ia and Ib). In addition to the lexicon, a LWFG has a set of constraint grammar rules. An example of a LWFG rule is given in Figure 1(II). Grammar nonterminals are augmented with pairs of strings and their semantic molecules. These pairs are called *syntagmas*, and are denoted by  $\sigma = (w, w') = (w, \binom{h}{b})$ . This rule generates the syntagma corresponding to *major damage* whose semantic molecule is given in Figure 1(Ic). There are two types of constraints at the grammar rule level — one for *semantic composition* (how the

meaning of a natural language expression is composed from the meaning of its parts) and one for *ontology-based semantic interpretation*. The composition constraints  $\Phi_c$  are applied to the heads of the semantic molecules, the bodies being just concatenated. Figure 1 shows that the body of the semantic molecule for *major damage* is a concatenation of the bodies of the adjective *major* and noun *damage*, together with a variable substitution. This variable substitution  $\{X_2/X, X_3/X\}$  is a result of  $\Phi_c$ , which is a system of equations — a simplified version of “path equations” (Shieber et al., 1983), because the heads are flat feature structures. These constraints are learned together with the grammar rules. The ontology-based constraints  $\Phi_o$  represent the validation on the ontology, and are applied to the body of the semantic molecule associated with the left-hand side non-terminal. The ontology-based interpretation is not done during the composition operation, but afterwards. Thus, for example, the head of the noun phrase *major damage* does not need to store the slot  $Y$ , a fact that allows us to use flat feature structures to represent the head of the semantic molecules. The ontology-based constraints are not learned; rather,  $\Phi_o$  is a general predicate applied to the logical form semantic representation which fully contains all the required information needed for validation on the ontology. Thus, it is independent of grammatical categories. This predicate can succeed or fail as a result of querying the ontology — when it succeeds, it instantiates the variables of the semantic representation with concepts/slots in the ontology ( $Y = degree$ ). For example, given the phrase *major damage*,  $\Phi_o$  succeeds and returns  $\langle X_1.isa = major, X.degree = X_1, X.isa = damage \rangle$ , while given *major birth* it fails.

### 3 Grammar Learning Algorithm

Unlike stochastic grammar learning for syntactic parsing (e.g., (Collins, 1999)), LWFG is well suited to learning from reduced-size training data. Furthermore, unlike previous formalisms used for deeper representations (e.g. HPSG, LFG), our LWFG formalism is characterized by a formal guarantee of polynomial learnability (Muresan, 2006).

A key to these properties is the partial ordering among grammar nonterminals, i.e., the set of nonterminals is well-founded. This partial ordering among nonterminals allows us to define the

*representative examples* of a LWFG, and to learn LWFGs from this small set of examples. The representative examples  $E_R$  of a LWFG,  $G$ , are the simplest syntagmas ground-derived by the grammar  $G$  — i.e., for each grammar rule, there exists a syntagma which is ground-derived from it in the minimum number of steps. Informally, representative examples are building blocks from which larger structures can be inferred via reference to a larger corpus  $E_\sigma$  which can be only weakly annotated (i.e., bracketed), or unannotated. This larger corpus,  $E_\sigma$ , is used for generalization during learning (Figure 2).

The theoretical learning model is *Grammar Approximation by Representative Sublanguage (GARS)* introduced in (Muresan, 2006; Muresan and Rambow, 2007). We proved that the search space for grammar induction is a complete grammar lattice, and we gave a learnability theorem for LWFG induction. The GARS model uses a polynomial algorithm for LWFG learning that takes advantage of the building blocks nature of representative examples. The learning algorithm belongs to the class of Inductive Logic Programming methods (ILP), based on entailment (Muggleton, 1995; Dzeroski, 2007). Unlike existing ILP methods that use randomly-selected examples, our algorithm learns from a set of representative examples allowing a polynomial efficiency for learning a syntactico-semantic constraint-based grammar, suitable to capture large fragments of natural language (Muresan, 2006).

The LWFG induction algorithm is a cover set algorithm, where at each step a new constraint grammar rule is learned from the current representative example,  $\sigma \in E_R$ . Then this rule is added to the grammar rule set. The process continues until all the representative examples are covered. We describe below the process of learning a grammar rule from the current representative example, illustrated as well in Figure 2.

**Step 1.** In the first step, the most specific grammar rule is generated from the current representative example. The category name annotated in the representative example gives the name of the left-hand-side nonterminal (“predicate invention”, in ILP terminology), while a robust parser returns the minimum number of chunks covering the representative example. The categories of the chunks give the nonterminals of the right-hand side of the most specific rule. For ex-

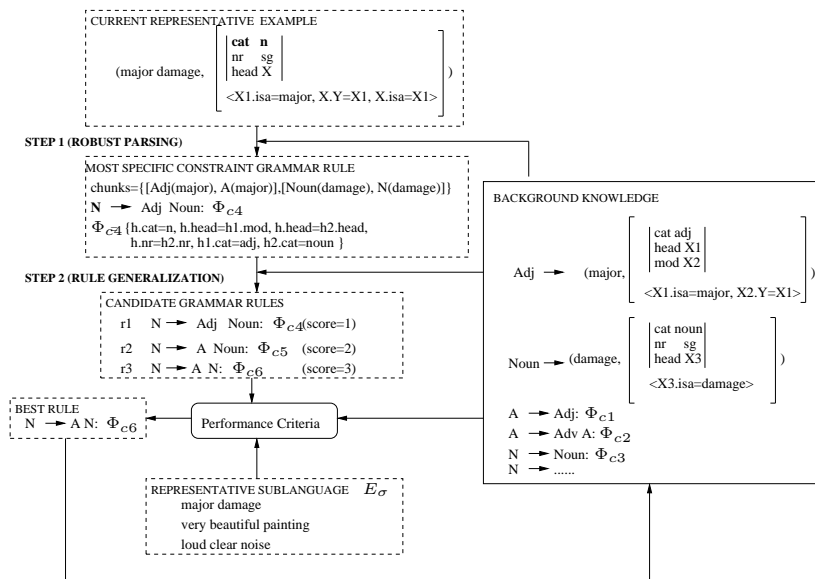


Figure 2: An iteration step of the learning algorithm

ample, in Figure 2, given the representative example *major damage* annotated with its semantic molecule, and the background knowledge containing the already learned rules  $A \rightarrow Adj$  and  $N \rightarrow Noun$ ,<sup>1</sup> the robust parser generates the chunks corresponding to the adjective *major* and the noun *damage*:  $[Adj(major), A(major)]$  and  $[Noun(damage), N(damage)]$ , respectively. The most specific rule generated is thus  $N \rightarrow Adj\ Noun: \Phi_{c4}$ , where the left hand side nonterminal is given by the category of the representative example, in this case *n*. The compositional constraints  $\Phi_{c4}$  are learned as well. It can be seen that the annotation of the representative example does not require us to provide ontology-specific roles or concepts. Thus, grammar learning is general, and can be done using a small, generic lexicon.

**Step 2.** In the second step, this most specific rule is generalized, obtaining a set of candidate grammar rules. The performance criterion in choosing the best grammar rule among these candidate hypotheses is the number of the examples in the representative sublanguage  $E_\sigma$  (generalization corpus) that can be parsed using the candidate grammar rule together with the previous learned rules. In Figure 2 given the representative sublanguage  $E_\sigma = \{major\ damage, loud\ clear\ noise, very\ beautiful\ painting\}$  the learner will generalize to the recursive rule  $N \rightarrow A\ N: \Phi_{c6}$ , since only this rule can parse

<sup>1</sup>For readability, we only show the context-free backbone of the grammar rules, and  $\Phi_o$  are not discussed since they are not learned.

all the examples in  $E_\sigma$ .

## 4 Levels of Representation

In order to transform natural language utterances to knowledge, we consider three levels of representation: the utterance level, the text level and the ontology level. In Section 4.4 we show that these levels of representation allow us to define meaning as  $Meaning = Text + all\ Questions/Answers$  w.r.t that *Text*, using a DAG matching approach.

### 4.1 Utterance-level Representation

At the *utterance level*, the semantic representation corresponds directly to a syntagma  $\sigma$  after the ontology constraint  $\Phi_o$  is applied. This representation is called Ontology-based Semantic Representation **OntoSeR**. At this level, the attrIDs are instantiated with values of the slots from the ontology, while the conceptIDs remain variables to allow further composition to take place. At OntoSeR level we can exploit the reversibility of the grammar, since this representation is used during parsing/generation.

In Figure 3 we show the semantic representation OntoSeR for the utterance *Hepatitis B is an acute viral hepatitis caused by a virus that tends to persist in the blood serum*, obtained using our parser in conjunction with our learned grammar. The composition constraints bind the conceptID variables, while the ontology constraint instantiates the attrID variables with values of slots in the ontology. The ontology constraint can be

*Hepatitis B is an acute viral hepatitis caused by a virus that tends to persist in the blood serum.*

**OntoSeR** =  $\langle (A.name=hepatitisB)_{HepatitisB}, (A.tense=pr)_{is}, (A.det=an)_{an}, (B.is\_a=acute, A.duration=B)_{acute}, (C.is\_a=viral, A.kind\_of=C)_{viral}, (A.is\_a=hepatitis)_{hepatitis}, (D.vft=ed, D.voice=pas, D.is\_a=cause, D.ag=E, D.th=A)_{caused}, (ag.is\_a=by, D.ag=E)_{by}, (E.det=a)_{a}, (E.is\_a=virus)_{virus}, (E.is\_a=that)_{that}, (F.tense=pr, F.is\_a=tend, F.no\_ag=E, F.prop=G)_{tends}, (G.vft=to, G.is\_a=persist, G.th=E)_{to\ persist}, (loc.is\_a=in, G.loc=H)_{in}, (H.det=the)_{the}, (I.is\_a=blood, H.of=I)_{blood}, (H.is\_a=serum)_{serum} \rangle$

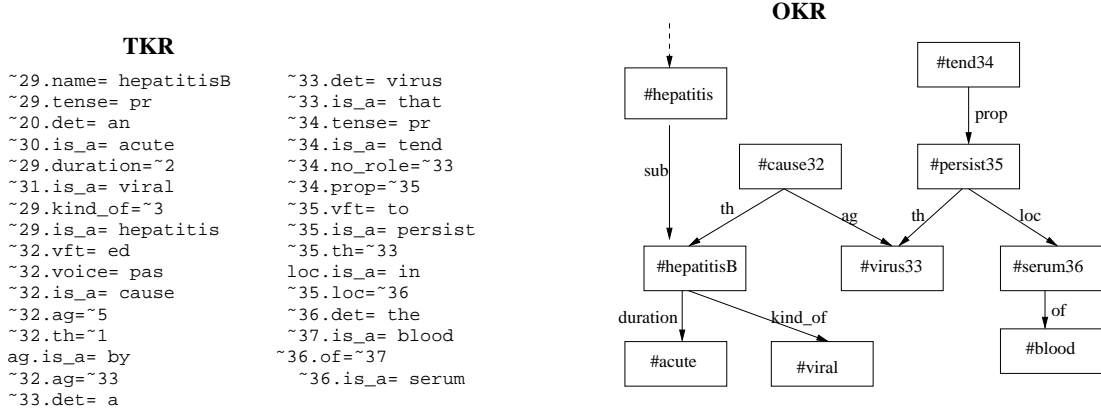


Figure 3: Example of an utterance and its levels of representation

seen as a *local semantic interpretation* at the utterance/grammar rule level, providing access to meaning during parsing/generation. In this paper, this semantic interpretation is based only on a weak “ontological model”. For the verb thematic roles we considered the thematic roles derived from Dorr’s LCS Database (e.g., *ag=agent*, *th=theme*, *prop=proposition*) (Dorr, 1997). For adjectives and adverbs we took the roles (properties) from WordNet (Miller, 1990). For prepositions we considered the LCS Database. We also have manually added specific/dummy semantic roles when they were not present in these resources (e.g., *of* between *blood* and *serum*).

The example in Figure 3 shows the output of our parser in conjunction with the learned grammar for a definitional sentence that contains several linguistic phenomena such as copula *to-be* predicative, reduced relative clauses (*caused by ...*), relative clauses (*virus that ...*), raising construction (*tends to persist*, where *virus* is not the argument of *tends* but the argument of *persist*), and noun compounds (*blood serum*). For readability, we indicate what part of OntoSeR corresponds to each lexical item. It can be noticed that OntoSeR contains representations of both ontological meaning (concepts and relations among concepts) as well as extra-ontological meaning such as tense and voice (*D.voice = pas*; *F.tense = pr*).

## 4.2 Text-level Representation

The *text-level* representation **TKR**, or discourse level representation, represents asserted representations. ConceptIDs become constants, and no composition can happen at this level. However, we still have (indirect) reversibility, since TKR represents all the asserted OntoSeRs. Therefore, all the information needed for reversibility is still present. Figure 3 shows an example of the TKR for the above utterance.

## 4.3 Ontology-level Representation

*Ontology-level* knowledge representation **OKR** is obtained after task-specific interpretation, which can be seen as a *global semantic interpretation*. OKR is a directed acyclic graph (DAG)  $G = (V, E)$ . Edges,  $E$ , are either semantic roles given by verbs, prepositions, adjectives and adverbs, or extra-ontological meaning properties, such as tense, aspect, modality, negation. Vertices,  $V$  are either concepts (corresponding to nouns, verbs, adjectives, adverbs, pronouns, cf. Quine’s criterion (Sowa, 1999, page 496)), or values of the extra-ontological properties such as **present** corresponding to *tense* property. In this paper, the task-specific interpretation is geared mainly towards terminological interpretation. We filter from OntoSeR determiners and some verb forms, such as tense, aspect, since temporal relations appear less in terminological knowledge than in factual

knowledge. However, we treat modals and negation, as they are relevant for terminological knowledge. An example of OKR for the above utterance is given in Figure 3.

We consider both concepts (e.g., #acute, #blood), and instances of concepts (e.g., #virus33, #cause32). Concepts are denoted in OKR by #name\_concept, and they form a hierarchy of concepts based on the *subsume* relation (*sub*), which is the inverse of the *is\_a* relation. An instance of a concept is denoted by the name of a concept followed by the instance number (e.g., #virus33). A concept and an instance of this concept are two different vertices in OKR, having the same name. At the OKR level we assume the *principle of concept identity* which means that there is a bijection between a vertex in OKR and a referent. For example, if we do not have pronoun resolution, the pronoun and the noun it refers to will be represented as two separate vertices in the graph. Currently, our semantic interpreter implements only a *weak concept identity principle* which facilitates structure sharing and inheritance.

To give these two properties we first introduce some notations. A DAG is called *rooted* at a vertex  $u \in V$ , if there exists a path from  $u$  to each vertex of the DAG. We have the following definition:

**Definition 1.** *Two subDAGs rooted at two vertices  $u, u'$  are equal if the set of the adjacent vertices to  $u$  and  $u'$  respectively, are equal and if the edges incident from  $u$  and  $u'$  have the same semantic roles as labels.*

**Property 1** (Structure Sharing). *In an OKR, all vertices  $u, u' \in V$  with the same name, and whose subDAGs are equal are identical (i.e., the same vertex in OKR).*

Using a hash table, there is a linear algorithm  $O(|V| + |E|)$  which transforms an OKR to an equivalent OKR which satisfies Property 1. In Figure 4 it can be seen that the OKRs of Hepatitis A and Hepatitis B share the representation corresponding to *blood serum* (i.e., *blood serum* is the same concept instance and due to Property 1 we have that #serum36=#serum27 and thus they have the same vertex in the OKR).

**Property 2** (Inheritance). *A concept in a hierarchy of concepts can be linked by the *sub* relation only to its parent(s), and not to any other ancestors. A subDAG defining a property of a concept from the hierarchy of concepts can be found only once in*

*the OKR at the level of the most general concept that has this property.*

For terminological knowledge we have that any instance of a concept is a concept, and the definition is the naming of a concept instance. For example, the definition of Hepatitis B, is an instance of a concept #hepatitis which has additional attributes *acute*, *viral* and *caused by a virus that tends to persist in the blood serum*. Thus, an additional instance of concept #hepatitis is created, which is named #hepatitisB. The fact that we can have the definition as a naming of a concept instance is facilitated also by our treatment of copula *to-be* at the OntoSeR level ( $A.name = hepatitisB, \dots, A.is\_a = hepatitis$  in Figure 3)

#### 4.4 Meaning as Answers to Questions

We consider that “understanding” a text is the ability to correctly answer, at the conceptual level, all the questions asked w.r.t to that text, and thus  $Meaning = Text + all\ Questions/Answers$  w.r.t that *Text*. In our framework we consider the principle of natural language as problem formulation, and not problem solving. Thus, we can represent at OKR level a paradox formulation in natural language, even if the reasoning about its solution cannot be emphasized. Our levels of representations allow us to define the meaning of questions, answers and utterances using a DAG matching approach.

**Definition 2.** *The meaning of a question,  $q$ , with respect to an utterance/discourse, is the set of all answers that can be directly obtained from that utterance/discourse. The semantic representation of a question is a subgraph of the utterance graph where the wh-word substitutes the answer concept(s).*

**Definition 3.** *The answer to a question is the concept that matches the wh-word through the DAG matching algorithm between the question’s subDAG and the utterance/discourse DAG.*

**Definition 4.** *The meaning of an utterance  $u$  is the set of all questions that can be asked w.r.t that utterance, together with their answers.*

Unlike meaning as truth conditions, where the problem of meaning equivalence is reduced to logical form equivalence, in our case meaning equivalence is reduced to semantic equivalence of DAGs/subDAGs which obey the concept identity principle (weak, or strong). The matching algo-

rithm obtains the same answers to questions, relative to semantic equivalent DAGs. If we consider only the weak concept identity principle given by Properties 1 and 2, the problem is reduced to DAG/subDAG identity.

## 5 Discussion

The grammar formalism, learning model and our ontology-based representation allow us to directly map text to graph-based meaning representations. Our method relies on a general grammar learning framework and a task-specific semantic interpreter. Learning is done based on annotated examples that do not contain ontology-specific roles or concepts as we saw in Section 3, and thus our learning framework is general. We can use any ontology, depending on the application. The task-specific semantic interpreter we are currently using is targeted for terminological knowledge, and uses a weak “ontological model” based on admissibility relations we can find at the level of lexical entries and a weak concept identity principle.

In (Muresan, 2006) we showed that our grammar formalism and induction model allow us to learn diverse and complex linguistic phenomena: complex noun phrases (e.g., noun compounds, nominalization), prepositional phrases, reduced relative clauses, finite and non-finite verbal constructions (including, tense, aspect, negation), coordination, copula *to be*, raising and control constructions, and rules for wh-questions (including long-distance dependencies).

In this section we discuss the processes of knowledge acquisition and natural language querying, by presenting an example of constructing terminological knowledge from definitions of *hepatitis*, *Hepatitis A* and *Hepatitis B*. The definitional text and OKRs are presented in Figure 4, OKR being shown only for the last two definitions for readability reasons. A question and answer related to the resulting OKR are also given.

The definiendum is always a concept, and it is part of the **sub** hierarchy. The concepts in the **sub** hierarchy are presented in bold in Figure 4. In addition to the concepts that are defined, we can also have *concepts that are referred* (i.e., they are part of the definiens), *if they do not have any modification* (e.g., **#blood** in definition of *Hepatitis A*, and *Hepatitis B*). If a referred concept has modifications, it is represented as an instance of a concept in OKR. As a consequence, various verbal-

izations of concept properties can be differentiated in OKR, allowing us to obtain direct answers that are specific to each verbalization. For example, the term *virus* appears in the definition of both *Hepatitis A* and *Hepatitis B*. In OKR, they are two different instances of a concept, **#virus25** and **#virus33**, since they have different modifications: *persists in the blood serum*, *does not persists in the blood serum*, respectively. These modifications are an essential part of the *differentia* of the two concepts **#hepatitisA** and **#hepatitisB**, causing the distinction between the two. When we ask the question *What is caused by a virus that persists in the blood serum?* we obtain only the correct answer **#hepatitisB** (Figure 4).

Another important aspect that shows the adequacy of our representation for direct acquisition and query is the OKR-equivalences that we obtain for different syntactic forms. They are related mainly to verbal constructions. Among OKR-equivalences we have: 1) active and passive constructions; 2) *-ed* and *-ing* verb forms in reduced relative clauses are equivalent to passive/active verbal constructions; 3) constructions involving raising verbs, where we can take advantage of the fact that the controller is not the semantic argument of the raising verb (e.g., in the definition of *Hepatitis B* we have *... caused by a virus that tends to persist in the blood serum*, while the question can be asked without the raising verb *What is caused by a virus that persists in the blood serum?*; see Figure 4).

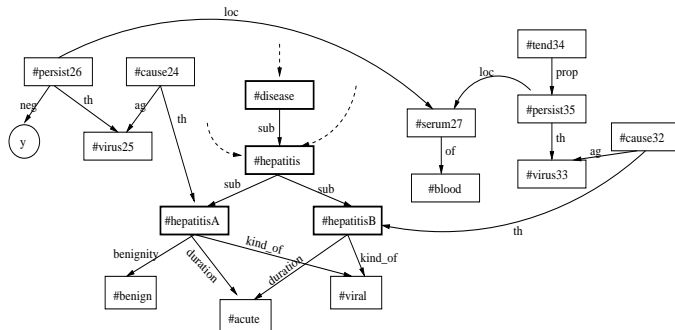
Besides acquisition of terminological knowledge, our grammar and semantic interpreter facilitates natural language querying of the acquired knowledge base, by treatment of wh-questions. Querying is a DAG matching problem, where the wh-word is matched to the answer concept.

## 6 Conclusions

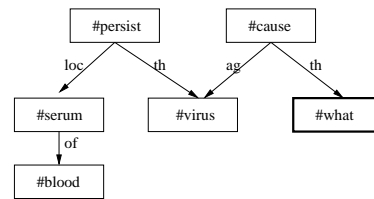
This paper has presented a learning framework to automatically map natural language to graph-based meaning representations via grammar induction. We presented an ontology-based semantic representation that allows us to define meaning as *Meaning=Text+all Questions/Answers w.r.t that Text*, using a DAG matching approach.

In the future, we plan to extend this work in two main directions. First, we plan to use a stronger semantic context with hierarchies of concepts and semantic roles, selectional restrictions, as well as

1. Hepatitis is a disease caused by infectious or toxic agents and characterized by jaundice, fever and liver enlargement.
2. Hepatitis A is an acute but benign viral hepatitis caused by a virus that does not persist in the blood serum.
3. Hepatitis B is an acute viral hepatitis caused by a virus that tends to persist in the blood serum.



**Q1:** What is caused by a virus that persists in the blood serum?



**A1: #hepatitisB**

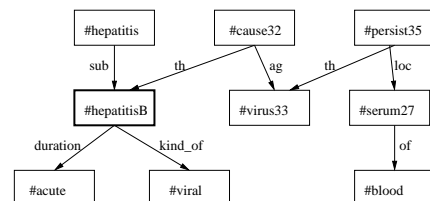


Figure 4: Acquisition/Query of terminological knowledge

semantic equivalences based on synonymy and anaphora. The second direction is to enhance the ontology with probabilities.

## References

- Basili, Roberto, Dorte H. Hansen, Patrizia Paggio, Maria Teresa Pazienza, and Fabio Zanzotto. 2004. Ontological resources and question answering. In *Workshop on Pragmatics of Question Answering, held jointly with NAACL 2004*.
- Beale, Stephen, Benoit Lavoie, Marjorie McShane, Sergei Nirenburg, and Tanya Korelsky. 2004. Question answering using ontological semantics. In *ACL 2004: Second Workshop on Text Meaning and Interpretation*.
- Bos, Johan, Stephen Clark, Mark Steedman, James R. Curran, and Julia Hockenmaier. 2004. Wide-coverage semantic representations from a CCG parser. In *Proceedings of COLING-04*.
- Collins, Michael. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.
- Copetake, Ann, Dan Flickinger, Ivan A. Sag, and Carl Pollard. 1999. Minimal Recursion Semantics: An introduction.
- Dorr, Bonnie J. 1997. Large-scale dictionary construction for foreign language tutoring and interlingual machine translation. *Machine Translation*, 12(4):271–322.
- Dzeroski, Saso. 2007. Inductive logic programming in a nutshell. In Getoor, Lise and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. The MIT Press.
- Ge, Ruifang and Raymond J. Mooney. 2005. A statistical semantic parser that integrates syntax and semantics. In *Proceedings of CoNLL-2005*.
- He, Yulan and Steve Young. 2006. Spoken language understanding using the hidden vector state model. *Speech Communication Special Issue on Spoken Language Understanding in Conversational Systems*, 48(3-4).
- Hovy, Eduard, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: The 90% solution. In *Proceedings of HLT-NAACL 2006*.
- Miller, George. 1990. WordNet: An on-line lexical database. *Journal of Lexicography*, 3(4):235–312.
- Muggleton, Stephen. 1995. Inverse Entailment and Progol. *New Generation Computing, Special Issue on Inductive Logic Programming*, 13(3-4):245–286.
- Muresan, Smaranda and Owen Rambow. 2007. Grammar approximation by representative sublanguage: A new model for language learning. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Muresan, Smaranda. 2006. Learning constraint-based grammars from representative examples: Theory and applications. Technical report, PhD Thesis, Columbia University.
- Nirenburg, Sergei and Victor Raskin. 2004. *Ontological Semantics*. MIT Press.
- Pereira, Fernando C. and David H.D Warren. 1980. Definite Clause Grammars for language analysis. *Artificial Intelligence*, 13:231–278.
- Shieber, Stuart, Hans Uszkoreit, Fernando Pereira, Jane Robinson, and Mabry Tyson. 1983. The formalism and implementation of PATR-II. In Grosz, Barbara J. and Mark Stickel, editors, *Research on Interactive Acquisition and Use of Knowledge*, pages 39–79. SRI International, Menlo Park, CA, November.
- Sowa, John F. 1999. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, CA.
- Wong, Yuk Wah and Raymond Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-2007)*.
- Zettlemoyer, Luke S. and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of UAI-05*.