Improving Semantic Integration By Learning Semantic Interpretation Rules

Michael Glass and Bruce Porter
Dept. of Computer Sciences
University of Texas at Austin
Austin, Texas, USA 78712-1188
{mrglass, porter}@cs.utexas.edu

Abstract

When extending a scientific knowledge base with new information, particularly information presented in natural language, it is important that the information be encoded in a form that is compatible with the existing knowledge base. Hand built systems for semantic interpretation and knowledge integration can suffer from brittleness. Methods for learning semantic interpretation and integration exist, but typically require large numbers of aligned training examples. Our approach to semantic integration learns rules mapping from syntactic forms to semantic forms using a knowledge base and a text corpus from the same domain.

A Framework for Scientific Knowledge Integration

The overall goal is to interpret and integrate natural language descriptions of science (chemistry, physics and biology) into a knowledge base. We assume an architecture with the following components: a syntactic parser, a non-empty knowledge base and a (possibly ambiguous) mapping from words in the domain to concepts in the knowledge base.

The semantic interpretation and integration proceeds in steps. First a syntactic parser processes a sentence. Then the resulting syntactic dependency tree is transformed into a logical form. The logical form is then integrated into the knowledge base using a representation that is consistent with the knowledge base. The resulting, augmented, knowledge base is then capable of drawing inferences or answering questions it could not previously answer. The goal of this research is to learn rules for transforming the syntactic parse into a logical form consistent with the existing knowledge base. We adapt research in the field of paraphrase acquisition to learn a mapping from syntactic paths to semantic paths.

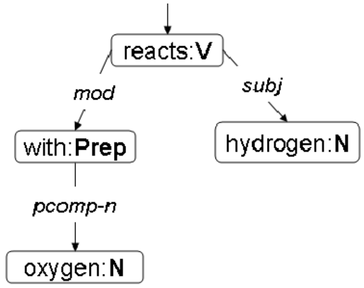
Previous Work

The method of semantic integration presented here builds on two lines of research: semantic interpretation and paraphrase learning. Direct Memory Access Parsing (DMAP) tightly integrates semantic interpretation with a knowledge base. Work on this method has used hand coded rules to map sentences to their interpretations. (Martin 1991) In other work, supervised learning algorithms processed sentences annotated with an interpretation to learn a model of the syntactic to semantic mapping, such as a synchronous grammar. (Wong & Mooney 2007)

Research in paraphrase acquisition has focused on two methods. The use of parallel, though not necessarily aligned, corpora is the most common. (Barzilay & McKeown 2001) More recently, distributional paraphrase acquisition has allowed paraphrase acquisition without parallel corpora.(Lin & Pantel 2001)

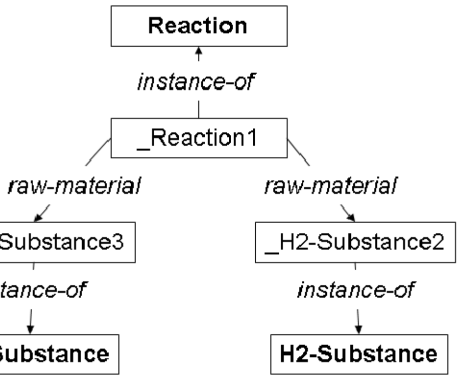
Parsing

The syntactic parser used in this research was Minipar (Lin 1993), though any parser capable of generating dependency trees could be used without significantly altering the algorithm. Although, it is significant that Minipar generates labeled dependencies. The key assumption is that syntactic dependencies will correspond to semantic dependencies.



Knowledge Representation

The method for improving knowledge integration described here assumes that there is an ontology and a significant existing knowledge base about the domain of interest, in this case chemistry, physics and biology. The knowledge base used in this research was The Component Library (Barker, Porter, & Clark 2001), though the method can be generalized to any knowledge representation system that can be viewed in a semantic graph form. The logical forms generated by the semantic interpretation and integration do not leverage the full representational power of KM (Clark & Porter 1998), they are simply graphs of instances.



Syntactic and Semantic Triples

Rather than representing the syntactic and semantic structures as graphs we use triples. Every triple (both syntactic and semantic) has the same basic form. There are two slot fillers that are the endpoints of the triple. These correspond to the words or concepts being related. There is also a path connecting these concepts: either a series of dependency relations and words for syntactic paths or a Component Library relation for semantic paths.

In the example above of the parsed sentence "Hydrogen reacts with oxygen" one syntactic triple is **reacts** $\mathbf{V} : \mathbf{subj} : \mathbf{N}$ **hydrogen**. The path is $\mathbf{V} : \mathbf{subj} : \mathbf{N}$ and the slot fillers are **reacts** and **hydrogen**. The corresponding semantic triple is **Reaction raw-material Hydrogen-Substance** whose path is **raw-material** and whose slot fillers are **Reaction** and **Hydrogen-Substance**.

A Baseline System

First we explain a baseline system that can be used in the presence of a parser, knowledge base and word to concept mapping. First the dependency tree is transformed into a simplified syntactic graph. The noun phrases and verbs are identified, these form the nodes in the syntactic graph. Then the paths between noun phrases and verbs are found, these make up the edges in the graph. The rest of the sentence is discarded. (See Figure 1) The word to concept mapping is then used to find a list of candidate concepts for each node, these are ranked according to their WordNet (Fellbaum 1998) sense number.

A candidate interpretation is defined as a semantic graph that is isomorphic to the syntactic graph and whose nodes are labeled with one of the candidate concepts from the corresponding syntactic node.(See Figure 2) The concepts on the nodes of the canidate are called the concept selections. The edges of the semantic graph are labeled with paths from the knowledge base, called the relation selections. Here we consider only paths of length one, a single relation. The likelihood of a candidate interpretation is the product of the likelihood for its concept selections and the likelihood of the relation selections.

The likelihood for any concept selection is related in a simple way to its WordNet sense number n (beginning with zero). Since more common senses are listed first, the likelihood for any particular concept selection is p^n. Where p is a parameter chosen with the property that each WordNet sense is approximately p times as common the WordNet sense before it. The likelihood for a set of concept selections is simply the product of the likelihood for each of them.

A relation selected between two concepts induces a triple. The likelihood for any relation selection is related to the most similar triple in the knowledge base. For any two triples their similarity is zero if their path is different and if the path is the same then the similarity is the product of their corresponding concept's similarities. The likelihood for a set of relation selections is the product of each relation selection.

The interpretation of a syntactic graph is chosen according to heuristics designed to select the candidate semantic graph with the maximum likelihood. The intuition behind this approach is that there are only so many ways two concepts can be related in a given knowledge base. Many pairs of concepts are not related at all. This constrains the concept selection. When selecting the relation between two concepts, if there is a relation between the concepts in the existing knowledge base that relation is selected. If there are multiple such relations one is chosen arbitrarily. If there are no such relations then two concepts are found that are most similar to the two given concepts such that the found concepts are related in the knowledge base. The relation between those concepts is chosen as the relation between the original concepts. So concept selection and relation selection are mutually constraining.

Limitations

The baseline approach has some obvious limitations. First, it depends on the correctness of the syntactic parse. Additionally, any part of the sentence other than noun phrases, verbs and paths between them is discarded. In particular, determiners such as **some** and **all** are discarded. The semantic graph will always be existentially quantified. There is also no attempt at anaphora resolution. These limitations will be retained in the refinement presented below. The only enhancement made here is that instead of unlabeled edges in the syntactic graph, the edges will be labeled with the syntactic path. The relation between nodes will be selected in a way that takes this path into consideration.

Extracting Training Data

In order to determine which syntactic paths are indicative of which semantic paths it is necessary to do some learning. A key advantage of our approach is that it requires no annotated data. Instead, we can extract syntactic triples from a corpus of scientific text and extract semantic triples from a knowledge base. The syntactic triples are extracted from the dependency trees of the parses and the semantic triples are extracted from the frames in the knowledge base.

Similarity Between Paths

Once each list of triples is extracted, the similarity of each pair of paths can be determined by the Extended Distribu-

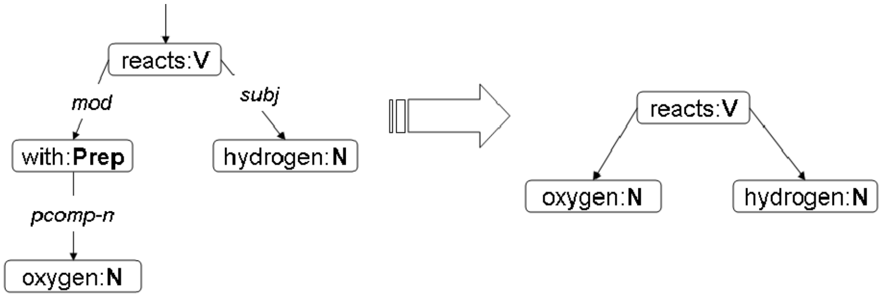


Figure 1: A syntactic graph from the dependency graph.

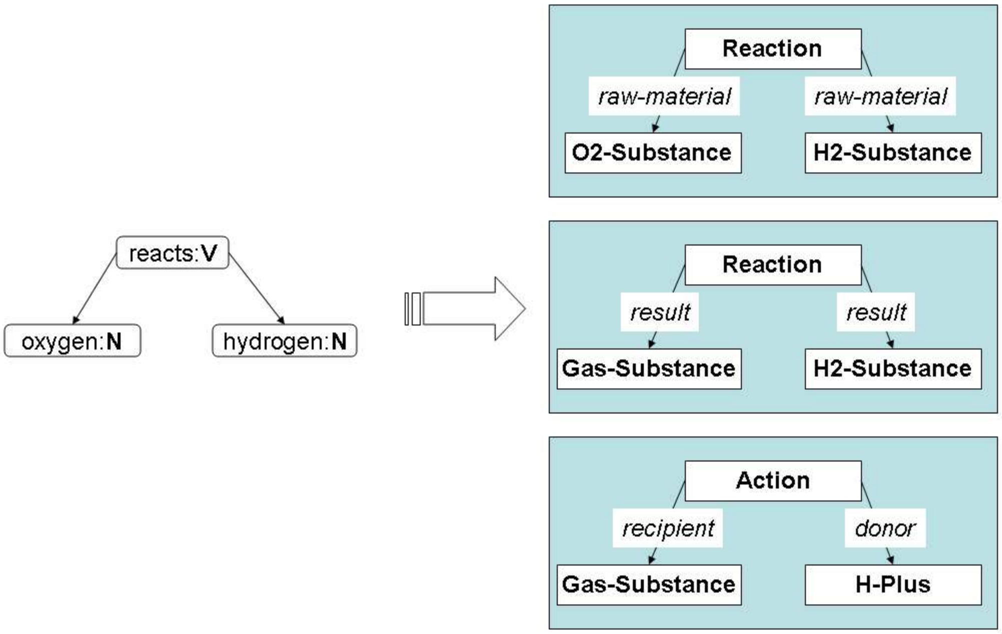


Figure 2: Candidate interpretations.

tional Hypothesis. This is an extension of the distributional hypothesis for words, that similar words occur in similar contexts. This hypothesis states that if two paths occur in similar contexts, the meanings of the paths tend to be similar. This hypothesis has found significant support in the work of Dekang Lin and Patrick Pantel whose system DIRT (Discovery of Inference Rules from Text) was able to learn paraphrases from a single corpus rather than a pair of parallel corpora.(Lin & Pantel 2001) The context of each path is the set of its slot fillers. So to find the semantic path most likely to be the interpretation for a particular syntactic path we can compare the set of slot fillers for the syntactic path to the set of slot fillers for each semantic path.

In the continuing example of "Hydrogen reacts with oxygen", both hydrogen and oxygen can be either **raw-material**s or **result**s of a reaction. In general, chemicals can be either reactants or products of reactions. So it is not obvious how a distributional learner can associate any syntactic path with one and not the other. The answer is that while some chemicals are common as both reactants and products others exhibit a bias to one or the other *in introductory chemistry texts*. Since both the knowledge base and the proposed syntactic corpus are drawn from this distribution, chemicals that are commonly given as reactants in example reactions will be common in both the **raw-material** slot filler and in syntactic paths suggestive of a reactant such as **Reaction** $V : subj : N$ **Thing**. Below is a table of the slot fillers more specific than **Chemical** in the triples matching **Reaction raw-material Chemical** and **Reaction result Chemical**.

Concept	**raw-material**	**result**
H2O-Substance	7	8
CO2-Substance	2	10
Ionic-Compound-Substance	3	5
H2-Substance	0	6
Ionic-Substance	0	6
Carbonate-Substance	3	2
NaNO3-Substance	0	4
Nitrate-Substance	3	1
O2-Substance	2	1
Metal	2	0
Chloride-Substance	0	2
Anion-Substance	0	2
Cation-Substance	0	2
NaOH-Substance	0	2

Although some chemicals, such as H20-Substance (water) and Ionic-Compound-Substance (a general concept for any ionic compound) are common as both reactants and products, some chemicals do exhibit a bias. If this bias is exhibited in the corpus of syntactic triples, the proper correspondences can be learned.

Word to Concept and Concept to Concept

Aside from the usual problems of finding a suitable similarity metric, there is the additional problem that syntactic fillers are noun phrases or verbs, while semantic fillers are concepts. Fortunately, there is a mapping from WordNet synsets to Component Library concepts. (Clark *et al.* 2005) The mapping of words to synsets is, however, many to many. So because of word sense ambiguity, the mapping of words to concepts is many to many. There is an additional complication in the form of the concept hierarchy. A pair of distinct concepts may be more similar than another pair of concepts. So to determine the similarity of syntactic fillers to semantic fillers it is necessary to first map the words to concepts and compare the concepts for similarity.

We use some simple methods for each task. We use a simple word sense disambiguation method based on choosing word senses that will allow an interpretation of the sentence consistent with the knowledge base. This is the same method used in the baseline system to select concepts for words. To determine the similarity of two distinct concepts we take the ratio of the number of shared superclasses to the total number of superclasses of each. If we consider the superclasses to be the features of a concept, this is the Jaccard similarity metric.

Learning Semantic Paraphrases

The conventional idea of paraphrases is to learn a path to path mapping. However, the syntactic paths can often be very general whereas the semantic paths are often specific to the particular concepts. For example in the sentence "A strong acid dissolves in water." the path from "strong acid" to "dissolves" is $\mathbf{V} : \mathbf{subj} : \mathbf{N}$ just as in the sentence "Hydrogen reacts with oxygen.". The proper semantic path in the case of dissolve is **object** but in the case of react it is **raw-material**. To address this difficulty we specialize the path to path mapping with a concept in one of the slot fillers. So rather than estimating the similarity of $\mathbf{V} : \mathbf{subj} : \mathbf{N}$ to **object** we estimate the similarity of **Dissolve** $\mathbf{V} : \mathbf{subj} : \mathbf{N}$ **Thing** to **Dissolve object Thing** and the similarity of **Thing** $\mathbf{V} : \mathbf{subj} : \mathbf{N}$ **Strong-Acid** to **Thing object Strong-Acid**. In order to avoid data sparsity we use the concept hierarchy to abstract the paths until they are general enough to have reliable samples.

The learned semantic paraphrases could be integrated into the baseline system to give an improved semantic interpretation and integration system. The paraphrases could also be used to assist another semantic interpretation system in selecting relations between concepts.

Conclusion and Experimental Design

The goal of this research is to learn a mapping from syntactic paths to semantic paths to aid in the intepretation of scientific text. This can be useful both for question answering and in integrating additional information into a knowledge base.

In order to evaluate this approach it will be necessary to construct a corpus of introductory level sentences in either the domain of chemistry, physics or biology. In theory, it would be possible to use the same textbooks the knowledge engineers used when constructing the knowledge base. However, the texts are too complex to even receive a even a correct parse on most sentences, and the word to concept assignment is even more challenging on complex sentences. Instead, a simplified version of English will be used to write the key facts from the textbook. These sentences can be accurately parsed and contain only the simplest cases of anaphora. Once the corpus is constructed it can serve to both train the semantic integration and to extend the knowledge base.

This model is to “pump prime” a knowledge base with basic information authored by knowledge engineers, then to allow a much more accessible knowledge authoring system, such as a simplified English, to extend it. A comparison of performance on question answering before and after the knowledge base is extended will evaluate the success of the knowledge integration system.

References

Barker, K.; Porter, B.; and Clark, P. 2001. A library of generic concepts for composing knowledge bases. In *Proceedings of the international conference on Knowledge capture*, 14–21. ACM Press.

Barzilay, R., and McKeown, K. 2001. Extracting paraphrases from a parallel corpus. In *Meeting of the Association for Computational Linguistics*, 50–57.

Clark, P., and Porter, B. 1998. KM - The Knowledge Machine: Reference manual. Technical report, University of Texas at Austin. http://www.cs.utexas.edu/users/mfkb/km.html.

Clark, P.; Harrison, P.; Jenkins, T.; Thompson, J.; and Wojcik, R. 2005. Acquiring and using world knowledge using a restricted subset of English. In *Proceedings of the 18th International FLAIRS Conference (FLAIRS’05)*.

Fellbaum, C. 1998. *WordNet – An Electronic Lexical Database*. MIT Press.

Lin, D., and Pantel, P. 2001. Dirt discovery of inference rules from text. In *Knowledge Discovery and Data Mining*, 323–328.

Lin, D. 1993. Principle-based parsing without overgeneration. In *ACL-93*, 112–120.

Martin, C. E. 1991. *Direct memory access parsing*. Ph.D. Dissertation, New Haven, CT, USA.

Wong, Y. W., and Mooney, R. J. 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)*, 960–967.