Proceedings of the ACL-02 Demonstrations Session, Philadelphia,
July 2002, pp. 116-117. Association for Computational Linguistics.

Knowledge Acquisition Incorporating Interactive NL Understanding

David Schneider, Kathy Panton, David Baxter, Jon Curtis, Pierluigi Miraglia, Nancy Salay
Cycorp, Inc.
3721 Executive Center Drive, Suite 100
Austin, Texas 78731
{daves, panton, baxter, jonc, miraglia, nancy}@cyc.com

Introduction A central issue in building knowledge-based systems is making them available to and understandable by naive users. This demonstration shows how a user and a knowledge-based system can collaborate to both create new knowledge and to extract existing knowledge from a knowledge base via natural language. KRAKEN[1] is an interface designed for users who have expert knowledge of some field, but no special training in knowledge representation, logic, or the like. Our basic assumption is that the interfacing medium between the user and the knowledge base should be as close as possible to natural language (though see (Clark et al. 2001) for another possible approach).

The demonstration will show how we parse from English into CycL (the logical representation used by the Cyc ontology and inference engine) using a variety of different parsing methods and post-processing steps, and will also demonstrate how this versatile representation language can be rendered back into English that is suitable for lightly trained users. These NLP tools are integrated with a large number of knowledge-engineering tools, since we (both Cycorp and the NLP community at large) have not succeeded in building a natural language system that can do everything necessary to construct knowledge via simple back-and-forth NL dialogue with a user.

Cyc's natural language understanding abilities consist of a lexicon with syntactic and semantic information, a hybrid top-down/bottom-up parsing system, a reformulation module, and a generation system.

Lexicon The lexicon (Burns and Davis 1999) contains syntactic, semantic, and pragmatic information for approximately 16,000 English root words. The lexicon also contains approximately 25,000 multi-word phrases and 25,000 proper names. Inflectional and derivational morphology are handled by a separate code component. Each root word is represented as a term in the knowledge base (KB), with assertions providing information about the word's part of speech, subcategorization patterns, and semantics. Semantic information in the lexicon involves a mapping between word senses and corresponding KB concepts or formulae.

Generation The natural language generation system produces a word-, phrase-, or sentence-level paraphrase of KB concepts, rules, and queries. The NLG system relies on information contained in the lexicon, and is driven by generation templates stored in the KB. These templates are not solely string-based; they contain linguistic data which allows, for example, for correct grammatical agreement to be generated. Semantic information in the templates is used to vary the generation depending on, for example, the semantic types of the arguments. The NLG system is capable of providing two levels of paraphrase, depending on the demands of the application. One type of generated text is terse but potentially ambiguous, and the other is precise but potentially wordy and stilted.

Parsing Our natural language understanding system parses input strings not simply into syntactic trees, but into fully-formed semantic formulas. Design criteria for the parsing system included that it (1) be fast; (2) produce parses of adequate semantic detail; (3) ask the user for clarification only in cases where the system could not itself resolve ambiguities; and (4) support parsing into underspecified formulas, and then rely on some of the other KRAKEN components to determine the best semantic translation.

The Text Processor controls the application of the various parsing subcomponents, using a heuristic best-first search mechanism that has information about the individual parsers, their applicability to coarse syntactic categories, cost, expected number of children, and so on. This information is used to perform a syntax-driven search over the parse space, applying relevant parsers to the subconstituents until all are resolved, or until the parsing options have been exhausted. The parsers at the disposal of the Text Processor are the Template parser, the Noun Compound parser, and the Phrase Structure parser.

[1] KRAKEN is being built as part of DARPA's Rapid Knowledge Formation Project (DARPA 2001).

The Template parser is essentially a top-down string-matching mechanism driven by a set of templates compiled into an efficient internal format. These templates employ a simple format so that users can add templates as they are entering new knowledge into the system. The template parser is relatively fast, but is of limited flexibility. It tabulates semantic constraints during a parse, but does not attempt to verify them; that task is passed along to the next processing layer.

The Noun Compound parser uses a set of semantic templates combined with a generic chart-parsing approach to construct representations for noun compounds such as “anthrax vaccine stockpile”. Unlike other parsing components, it makes heavy use of the knowledge base, and can therefore resolve many ambiguities that are impossible to handle on a purely syntactic level (e.g. “Mozart symphonies” vs. “Mozart expert”).

The Phrase Structure parser takes a similar bottom-up approach to constructing parses. After completing a syntactic parse, it uses semantic constraints gleaned from the KB to perform pruning and to build the semantic representation.

Post-Processing In order for parsing to be successful in the current application, some decisions about semantic meaning need to be deferred during parsing. In particular, radically vague or underspecified words such as ‘is’ or ‘contains’, which can map onto many distinct relations in the KB, introduce ambiguities which are not handled well by producing all possible interpretations in parallel. To deal with such cases, strings are parsed into an intermediate layer (called iCycL) that conflates relevant ambiguities into a single parse, by using very general predicates such as is-Underspecified. The Reformulator reformulates iCycL representations into final, more specific CycL representations, often with the user’s help.

In addition to handling underspecification, the iCycL layer is also well-suited for other types of semantic processing, such as interpretation of quantification and negation, and type-shifting. The interpretation of quantifiers, for example, occurs as a transformation from iCycL expressions into full-fledged CycL logical forms. Although CycL representations are modeled on first-order logic, the language itself allows the definition of higher-order constants. We exploit this capability to represent a wide range of NL quantifiers formally as generalized quantifiers, i.e., as higher-order relations between sets of objects.

Lexical Additions Because KRAKEN users are allowed to add new terms to the ontology, we have created a special-purpose tool to elicit the information necessary to be able to parse to and generate from new terms in the ontology. The Dictionary Assistant allows the user to specify appropriate syntactic and semantic information through a dialogue in which they do things like verify the part of speech (count noun, adjective, etc.) that the system has assigned, declare it to be any of a large number of types of names, and tell the system whether or not this is the preferred way to refer to the new term. For new relations, the user can add both parsing and generation templates.

The Knowledge Base The KB we use is currently the largest knowledge base in the world, housing over 1.4 million largely hand-entered facts and rules that interrelate more than 100,000 concepts. Concepts are denoted in the KB with constants; these may be individuals, intensionally defined or extensionally defined collections, or relations which obtain between terms. Functions can be used to refer to many more individuals without reifying a separate term for each (e.g. the concept “gander” is represented virtually in the KB by the nonatomic term *(MaleFn Goose)*). Currently, the KB has knowledge of a wide range of topics, including things as diverse as microbiology and pop music, and has an extensive knowledge infrastructure, including multiple treatments of causality and temporal and modal reasoning. Knowledge is clustered into context-specific domains (or “microtheories”) with epistemic access determined by specialized predicates. Consequently, the system has an ability to differentiate between and accommodate logically conflicting bodies of knowledge, including hypothetical and counterfactual contexts.

Acknowledgements The authors are indebted to the work and contributions of our company’s Knowledge Formation and Dialogue group: Michael Witbrock, Bjørn Aldag, Stefano Bertolo, Keith Goolsbey, Fred Hoyt, John Jantos, Robert Kahlert, Kim Loika, Matthew Olken, Roland Reagan, Matt Smith, Jennifer Sullivan, and Peter Wagner.

References

Burns, K. J. and Davis, A. R. 1999. Building and maintaining a semantically adequate lexicon using Cyc. In Viegas, E. ed. *Breadth and depth of semantic lexicons*. Dordrecht: Kluwer.

Clark, P.; Thompson, J.; Barker, K.; Porter, B.; Chaudhri, V.; Rodriguez, A.; Thomere, J.; Mishra, S.; Gil, Y.; Hayes, P.; Reichherzer, T. 2001. Knowledge Entry as the Graphical Assembly of Components. In Proceedings of the 1st International Conference on Knowledge Capture (K-Cap’01), 2001.

DARPA. The Rapid Knowledge-Formation Project (website). http://reliant.teknowledge.com/RKF/, 2001.