Deep Sentence Understanding in a Restricted Domain*

Pierre Zweigenbaum and Marc Cavazza
DIAM — INSERM U.194, 91 bd. de l'Hôpital, 75634 Paris Cedex 13, France
e-mail: zweig@frsim51 (bitnet)

Abstract

We present here the current prototype of the text understanding system HÉLÈNE. The objective of this system is to achieve a deep understanding of small reports dealing with a restricted domain. Sentence understanding builds a model of the state of the world described, through the application of several knowledge modules: (i) LFG parsing, (ii) syntactic disambiguation based on lexical entry semantic components, (iii) assembly of semantic components and instantiation of domain entities, and (iv) construction of a world model through activation of common sense and domain knowledge.

1 Introduction

We present here the current prototype of the text understanding system HÉLÈNE. The objective of this system is to achieve a deep understanding of small reports dealing with a restricted domain (here, patient discharge summaries in a medical specialty). This means that HÉLÈNE should rely on an extensive description of all types of required knowledge. This implies of course a deep domain knowledge base, including the needed common sense knowledge.

Precise understanding must not only rely on complete domain knowledge, but also on enough syntactic information. This is the reason why HÉLÈNE includes a full syntactic module, whose task is to provide the semantic construction module with the (deep) structures of sentences. One problem with syntactic processing is that it gives rise to numerous ambiguities. These ambiguities are filtered on semantic grounds by a disambiguation module that does not build any semantic representation.

Semantic construction is concerned with the recognition of domain entities that can be expressed by word groups. We thus had to adopt a lexical semantics approach compatible with descriptions. Domain entities, once instantiated, provide the basis on which a model of the current state of the world (here, the patient state) is built. The same lexical semantic information is used both to help syntactic processing, and in a more extensive way to access domain models in order to build semantic representations.

The prototype includes the following main modules:

- The syntactic module implements the Lexical Functional Grammar formalism [7]. The parser builds c-structure and f-structure bottom-up in parallel on a chart, so that f-structure validity can constrain c-structure construction.
- Ambiguous attachments are submitted to evaluation and ranking by the disambiguation module. This module applies a set of general heuristic rules that operate on the semantic definition of the LFG predicates.
- Semantic construction relies on dynamic domain models that integrate common sense. LFG predicates are characterized by semantic components that point to parts of the knowledge base.

The prototype runs in Common Lisp (VAX Lisp) and K, a proprietary language embedded in Common Lisp. The remaining sections describe these modules in more detail.

2 Parsing with a Lexical-Functional Grammar

We chose to implement the LFG framework for several reasons. Being a linguistic theory,

*supported by AIM project 1003 and PRC Communication Homme-Machine, Pôle Langage Naturel.

it should provide better foundations for principled syntactic coverage. A formalism with a context-free backbone was easier to implement. Furthermore, LFG extracts from a sentence a predicate-argument structure which consitutes a good starting point for semantic processing. Our implementation of LFG does not include yet a schema for long-distance dependencies (or functional uncertainty) and coordination. It allows cyclic f-structures.

Our parser uses a chart to build both c-structure and f-structure. Incomplete and complete constituents are represented by active and inactive cs-edges, while incomplete and complete f-structures are placed on active and inactive fs-edges. The parsing strategy is bottom-up, left to right (left corner). Top-down parsing is also available, as well as right to left parsing. LFG schemas are evaluated as soon as possible. Equational (construction) schemas are evaluated when encountered, and constraint schemas (existential, equational and negation of those) are kept on fs-edges until they can be evaluated. When fs-edges are combined, remaining constraints are propagated to the resulting fs-edge. Each new active f-structure is tested for consistency and coherence. Furthermore, the value of a closed function is tested for completeness (this should be revised if a scheme for long-distance dependencies is implemented). When a constraint is violated, its fs-edge is flagged as invalid.

Grammar rules are described as regular expressions which are compiled into (reversible) transition networks. Each arc of those networks is labelled with a category and a disjunction of conjunctions of schemas. A model of hierarchical lexical entry representation has been developed, with data-driven lexical rules. It is not currently coupled to the parser, and will not be presented here. The prototype uses a simple word list with what would be the result of this model as lexical entries.

The prototype uses a small French grammar that contains 14 networks, equivalent to 90 rules. It was assembled by borrowing from the literature [3,10] and by completing with grammar manuals and real text confrontation. It has the particularity of building cyclic f-structures for constructions where a head plays a role inside an adjunct. This is how we process attributive adjectives, participial phrases, and (in a very limited way) relative phrases.

3 Semantic rules for Syntactic Disambiguation

Structural ambiguity is ubiquitous in our target texts, since they contain descriptions that often make use of series of prepositional phrases to qualify a noun. We have then decided to submit ambiguous attachments to semantic approval and ranking before building complete parses.

An ultimate test of semantic validity would consist in comparing complete semantic representations built for each attachment proposal [1]. However, such a method is too expensive to allow systematic application. Our system implements a more tractable approach that generalizes selectional restrictions (or preferences). Evaluation is performed by executing a set of heuristic positive and negative rules that vote for or against each proposal. Rule conditions embody criteria that refer to the semantic components (see below) of the predicates to be attached, and include the notion of isotopy [8]. They apply not only to predicate–argument selection, but also to predicate–adjunct combination.

4 Semantic Construction

Semantic processing of a sentence results in the activation of a relevant body of domain knowledge with related inferences within the knowledge base.

Domain knowledge (here concerning a single disease: thyroid cancer) is embedded in a model [6,4] describing domain objects, actions operating on them and specific processes involving these objects. Such a model is thus a dynamic causal model rather than a memory structure devoted to object and event integration [9]. It is analogous to deep-knowledge models used in modern expert systems [2].

Domain objects are represented in a frame-like formalism. Actions and operative aspects of processes are described as production rules simulating a distributed parallel activation [4]. The whole model corresponds to a dynamic, data-driven environment.

Some domain concepts specifically represent states, relationships between objects, or state transitions. They can be triggered by their occurrence as word meanings in the sentence. Implicit occurrence of these concepts may also be recognized by observing the evolution of the model. In this case default procedures create the corresponding concepts inside the model just as if these elements were explicitely stated in the proposition. These concepts subsume important situations in the model and translate them into a higher description level, thus allowing output to the user for a trace of correct understanding.

No deep understanding would be possible without a treatment, even partial, of common sense, which in this application is concerned mainly with part-whole relationships [5], reasoning about transitions and change, and elementary physical actions (e.g., removing, touching). Default knowledge on actions, roles and reference (e.g., used in the resolution of pragmatic anaphora) are associated to the common sense module.

Common sense mechanisms are incorporated as production systems similar to those describing other active elements of the model, and can thus recombine freely with them in order to complete or modify existing representations.

Domain representations are built from the assembly of lexical contents along the syntactic structure of the proposition. Words contain semantic components [8], which are markers referring to elements of the knowledge base or properties of these representations. The existence of explicit common sense concepts in the knowledge base makes it possible to decompose homogeneously technical and ordinary words.

The lexical contents are assembled by heuristic rules to form candidate domain objects which are recognized as instances of prototypes in the representation. Lexical content itself is loosely structured; the association of the components is completed according to their type (which is derived from the type of entity they refer to) and the dependency (predicate-argument, predicate-adjunct) relations between the lexemes that contain such components, as provided by the LFG.

As such elements of representation are recognized by the model, the reactive environment is triggered and interprets data until new information is analyzed.

The prototype currently runs on a small set of 30 sentences taken from patient discharge summaries. These sentences were selected for the linguistic issues they illustrate and the domain inferences they trigger. A fully compiled version of the program running on a VAX 8810 processes a sentence in an average 12 sec. CPU time.

References

[1] Birnbaum, L. (1985). Lexical ambiguity as a touchstone for theories of language analysis. IJCAI 9, Los Angeles, 815–820.

[2] Chandrasekaran, B. and Mittal, S. (1983). On deep versus compiled knowledge approaches to medical diagnosis. *Int J Man-Machine Studies* 19:425–436.

[3] Grimshaw, J. (1982). On the lexical representation of Romance reflexive clitics, in Bresnan (ed.), *The Mental Representation of Grammatical Relations*, Cambridge, Mass., MIT Press.

[4] Holland, J.H., Holyoak, K.J., Nisbett, R.E., Thagard, P.R. (1986). *Induction: Processes of Inference, Learning, and Discovery.* Cambridge: MIT Press.

[5] Iris, M.A., Litowitz, B.E., and Evens, M. (1988). Problems of the part-whole relation. In Evens, M.E., ed. *Relational Models of the lexicon.* Cambridge University Press.

[6] Johson-Laird, P.N. (1983). *Mental Models*, Cambridge University Press.

[7] Kaplan, R.M., and Bresnan, J. (1982). Lexical-Functional Grammar: A Formal System for Grammatical Representation, in Bresnan (ed.), *The Mental Representation of Grammatical Relations*, Cambridge, Mass., MIT Press.

[8] Rastier, F. (1987). *Sémantique Interprétative.* Paris: Presses Universitaires de France.

[9] Schank, R.C., and Abelson, R.P. (1977). *Scripts, Plans, Goals and Understanding.* Lawrence Erlbaum Associates.

[10] Waite, J.J. (1986). Grammaire Lexicale-Fonctionnelle — Dislocation, inversion stylistique et constructions impersonnelles. PhD Thesis, University of Montreal.