Realization and Directed Parsing

Rolf Backofen and John Nerbonne
Deutsches Forschungszentrum für Künstliche Intelligenz
ASL and DISCO Projects
Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, Germany
{backofen,nerbonne}@dfki.uni-sb.de

Précis: Among the kinds of expertise relevant for successful natural language understanding are grammar, semantics and background knowledge, all of which must be represented in order to decode messages from text (or speech). In this extended abstract we examine sortal disambiguation—the choice of or directed search for one of several hypotheses about the meaning of an input string based on sortal information—as an example where modules need to cooperate. We demonstrate how to provide for a cooperation of background knowledge (represented in taxonomic logic) with grammar representations. The integration of taxonomic and grammatical information has the advantage of allowing a very close integration of background information with grammar—and therefore an early elimination of some analysis hypotheses. We argue here further that an effective use of taxonomic information for the purpose of directing parsing involves the realization facility of taxonomic logic.

Disambiguation

DISAMBIGUATION is the process of determining (i) which of potentially many meanings was intended in an utterance, but also (ii), with respect to a particular application, which facet is relevant to an NL interaction. The former is a response to the ambiguity of natural language, while the latter exists even where no genuine ambiguity does. We illustrate these in turn below. Disambiguation is extremely important in applications in which there many be uncertainty about input—e.g., speech. Like parsing itself, at least some disambiguation seems to be automatic, so that untrained speakers are not aware of needing to disambiguate structures. The example below, graphed in Figure 1, suggests how unobtrusive the process is:

(1) a. Who bought books in Spanish?
    b. Who bought books in May?

This sort of example is convenient because it shows how pervasive the effects of disambiguation may be—reaching even into the parsing component. It is simultaneous misleading if it suggests that genuine disambiguation tasks need to be accompanied by such striking consequences. For even if disambiguation MAY be accompanied by striking consequences in application independent ways, the need for disambiguation arises in NLP interface efforts in ways that need have no purely linguistic ramification whatsoever. In particular, NL interfaces need to be sensitive to application distinctions which do not correspond to natural language ambiguities.

Consider an early DISCO application, that of consulting with multiple agents who plan shipping. Here the phrase *Schmidts Ladung* 'Schmidt's freight' certainly denotes freight which stands in some relation to Schmidt. For example, we may imagine the freight contracts in the application as organized into a small database, where the freight contract is the basic tuple.

Order Nr.	Contractor	Agent	Destination	?
457	Schmidt	...	...	...
574	...	Schmidt	...	...
745	...	...	Schmidt	...
475	...	...	...	Schmidt

Thus the phrase *Schmidts Ladung* could designate freight which Schmidt contracted to have shipped, freight for which he is the freight agent, freight being sent to him, and perhaps even freight which stands

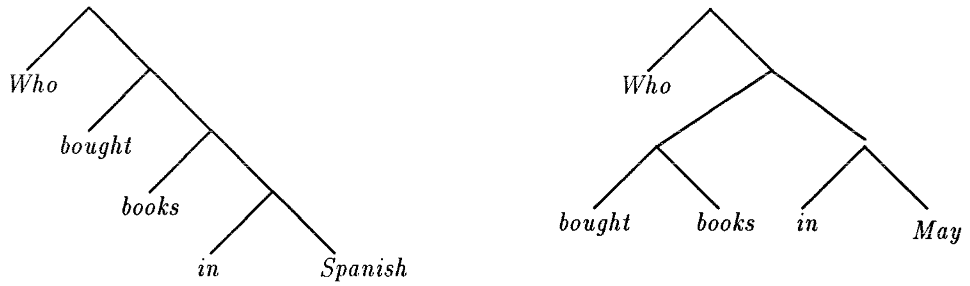

Figure 1: Disambiguation may even require the recognition of distinct constituent structures. Note that the proper names *Spanish* and *May* are not syntactically distinct, nor do that belong to distinct logical types—each denotes an entity. But they do denote objects of different SORTS, as this term is used in sortal logic or taxonomic logic, since May is a time and Spanish a language. The link to knowledge representation, especially of the kind encoded in KL-ONE, is justified by the emphasis on sorts—taxonomies, or conceptual hierarchies—which distinguishes knowledge representation schemes such as KL-ONE.

in yet another relation to him (as owner, inspector, as packer, etc.). Now it is unlikely that the relation expressed by the German possessive construction (genitive + $\bar{\text{N}}$) is ambiguous, and it is unthinkable that the construction is grammatically ambiguous to just this degree and in just this fashion.

Taxonomic reasoning of the KL-ONE variety (Brachman and Schmolze 1985, Baader and Hollunder 1990) may fruitfully be applied both to the resolution of linguistic ambiguity and to resolution of application-specific distinctions (Bobrow 1979). We consider these in turn. At the heart of taxonomic reasoning is the imposition of a sort hierarchy on the domain, illustrated in Figure 2 for the case where we found linguistic ambiguity.

In addition to the provision of a sort hierarchy, sortal disambiguation requires a characterization of which sorts are appropriate for which (argument positions of) relations. We would then allow that *in* translate into (at least) two relations, one temporally relating eventualities to times, and the other relating documents to media (but not to times). Schematically:

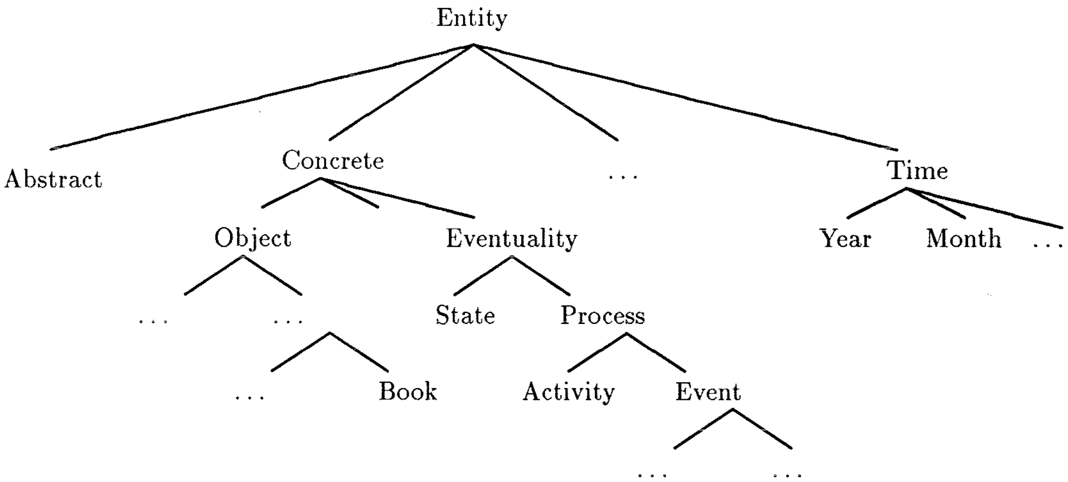

Figure 2: A sort hierarchy which distinguishes enough classes in the domain to illustrate sortal disambiguation for the sentences in (1) (in the text). The hierarchy might better be extended from a tree to a directed graph—in which sorts inherit on more than one path back to the root. For example, states and activities might best be viewed as subsorts of both eventualities and nondiscretes, which would include physical substances (water, flour) as well. The move to non-tree-structured hierarchies does not affect the points here however, about the use of taxonomic reasoning to direct parsing.

Lexical Item	relation	Argument1-Sort	Argument2-Sort
in	temporal-loc	Eventuality	Time
	use-medium	Document	Medium

Finally, we must enforce the sortal compatibility restrictions. For many applications, it is desirable to enforce these as early as possible, so that unnecessary processing is avoided. Of course, this mechanism takes us outside KL-ONE proper, but the required information—compatibility—is efficiently provided in KL-ONE-like systems. As the example in Figure 1 suggests, an enforcement of sortal compatibility as early as parse time would be useful (and recall that, e.g., speech applications will rely on disambiguation to prune unlikely hypotheses). This raises the question of how well these constraints can be integrated into other processing—which of course depends on whether they can be expressed in the formalisms of other modules. Here then is a concrete instance of the question of how one relates knowledge representation to grammatical and semantic formalisms.

Earlier work in DISCO demonstrates that semantics can be formulated in an indirect fashion in feature formalisms, so we shall show here that the same is true of knowledge representation—at least within bounds. Cf. Moens et al. 1989 for an earlier proposal along the same lines. That is, once we've taken the step of representing the semantics of *in* in a typed feature description language:

$$\begin{bmatrix} \text{PRED} & \textbf{temporal-location} \\ \text{THEME} & \\ \text{LOCATION} & \end{bmatrix}$$

then we can also represent the sortal information, relying on unification to enforce sortal compatibility, and thus integrating sortal disambiguation with the unification used in parsing. The following feature structure description represents the ambiguous lexical item *in*:

$$\begin{bmatrix} \text{FORM} & \text{in} \\ \text{SEMANTICS} & \left\{ \begin{matrix} \begin{bmatrix} \text{PRED} & \textbf{temporal-location} \\ \text{THEME} & [\ \textit{Eventuality}\ ] \\ \text{LOCATION} & [\ \textit{Time}\ ] \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \text{PRED} & \textbf{expressive-means} \\ \text{THEME} & [\ \textit{Document}\ ] \\ \text{LOCATION} & [\ \textit{Medium}\ ] \end{bmatrix} \end{matrix} \right\} \end{bmatrix}$$

The representation for the word *May*, whose semantics is of the sort Month, and therefore also of the sort Time, can successfully unify with the (location argument of) the first alternative semantics for *in*, but not the last, for which an argument of the sort Medium is expected. Thus the PP *in May* seeks to attach where its first argument will be of the sort Eventuality—and this can be a VP attachment, since VP's denote eventualities, but not an NP with the head noun *books*, since this denotes objects of an incompatible sort.

Although we shall not present the details of the treatment of the resolution of application-specific distinctions, it should be clear that the same techniques apply. In the example *Schmidts Ladung*, the relation between Schmidt and the freight is potentially disambiguated by information about whether Schmidt is a shipper, a customer, or the recipient of a customer's shipment. Nor shall we attempt on the basis of this example to argue that sortal restrictions must come from the domain and NOT from the lexicon—the dilemma seems spurious, since the lexicon must in some way be accommodated to the domain for serious applications anyway. Cf. Iida et al. 1989 on the relation between lexicon and disambiguation in complex applications.

Realization

There are two ways of employing this information in feature logics. The first one is to encode in a disjunctive feature structure all possible semantic values (cf. above), and to maintain consistency of the feature structure during all processing steps. But this method, relying as it does on disjunction, is computationally inefficient and it merely filters incorrect hypotheses. We obtain predictive power for directing parsing and the stochastic search process of speech, but it comes at the expense of exponential update costs.

The second way of employing the information would resemble KL-ONE realization. The idea is to include only partial semantic information within the feature structure, and to maintain an additional set of conditional constraints about sortal compatibility. This information is not updated throughout processing, but only when an antecedent is satisfied (subsumed), so that update costs grow polynomially. When an antecedent "fires", e.g., when an argument is bound to an argument position, one then checks, using realization, for a characterization of (an appropriate supersort of) semantics. In this case the supersort information may be added, thus serving as additional information for the processing. Note that this information characterizes the space of remaining hypotheses—so that it allows sortal information to actively direct parsing or stochastic search.

This kind of realization could be accomplished by adding implications of form $A_1 \rightarrow B_1 \ldots A_n \rightarrow B_n$ to a semantic entry C. Here the antecedents $A_1 \ldots A_n$ are feature structures with the property that if C is subsumed by A_i, then C is of sort S_i. B_i is then the definition of the sort S_i in the sort hierarchy. Newer work (Aït-Kaci et al. 1992) shows that checking subsumption *and* non-subsumption (or entailment as subsumption is called in this framework) can be done very efficiently.

Now in the worst case A_i is just B_i of sort S_i—no information is gained. Furthermore it could happen that at the end of the syntactic processing none of the implications will have failed or fired. Then we would have to unify C with the disjunction $\{B_1, \ldots, B_n\}$ in order to check consistency. In this (the worst) case we would not have won any efficiency. But in the case of a semantics with relatively elaborate sortal restrictions there could be an appropriate hierarchy that allows one to avoid the worst case. We examine this now in an attempt to characterize when sortal information is useful.

So let $B = \{B_1, \ldots, B_n\}$ be the disjunction of all possible semantics entries. Furthermore we assume that there are feature structures $A_1, \ldots, A_n$ that partition a domain A such that

1. $B_i \sqsubseteq A_i$
2. $A_i \sqcap A_j = \perp$ for $i \neq j$,
3. $A_1 \sqcup \ldots \sqcup A_n = A$.

The second condition implies $\neg A_j \sqcap A_i$ is equivalent to A_i which implies that $\neg A_j \sqcap B_i$ is equivalent to B_i. Then it is easy to check that the conjunction $A_1 \rightarrow B_1 \sqcap \ldots \sqcap A_n \rightarrow B_n$ is equivalent to the disjunction $\{B_1, \ldots, B_n\}$. To see this examine the case $n = 2$. Then

$$\begin{aligned}(A_1 \Rightarrow B_1) \sqcap (A_2 \Rightarrow B_2) &= (\neg A_1 \sqcup B_1) \sqcap (\neg A_2 \sqcup B_2) \\ &= (\neg A_1 \sqcap \neg A_2) \sqcup (\neg A_1 \sqcap B_2) \sqcup (\neg A_2 \sqcap B1) \sqcup (B_1 \sqcap B_2)\end{aligned}$$

Given what is said above this simplifies to $B_1 \sqcup B_2 \sqcup (B_1 \sqcap B_2)$, which is indeed the same as $\{B_1, B_2\}$.

This means that if we can organize semantic hypotheses into a set partitioning a domain, they should find effective deployment in disambiguation and directed parsing.

Summary and Conclusions

We have investigated the use of taxonomic information for the purpose of disambiguation and directing parsing and the expression of taxonomic information in feature logics. We have argued that taxonomic reasoning is clearly useful in disambiguation, and that the realizer facility of taxonomic logics is essential for predictive disambiguation. We have also shown how most of the taxonomic reasoning important for disambiguation may be implemented in feature logics, allowing an integration of taxonomic reasoning in feature logics, and that the classifier is definable in feature logics. Nerbonne 1992 discusses some potential limits on the extent to which feature logics are suitable for the expression of disambiguating information, a topic of ongoing research. An implementation of the integrated concept with classifier awaits further work.

References

Aït-Kaci, H., A. Podelski, and G. Smolka. 1992. A Feature-based Constraint System for Logic Programming with Entailment. In *Fifth Generation Computer Systems 1992*, 1012–1021. Tokyo, Japan, June. Institute for New Generation Computer Technology.

Baader, F., and B. Hollunder. 1990. KRIS: Knowledge Representation and Inference System—System Description. Technical memo, Deutsches Forschungszentrum für Künstliche Intelligenz, Kaiserslautern.

Bobrow, R. 1979. Semantic Interpretation in PSI-KLONE. Annual Report 4274, Bolt, Beranek and Newman, Cambridge, MA.

Brachman, R. J., and J. G. Schmolze. 1985. An overview of the KL-ONE knowledge representation system. *Cognitive Science* 9:171–216.

Iida, M., J. Nerbonne, D. Proudian, and D. Roberts. 1989. Accommodating Complex Applications. In *Proceedings of the First International Workshop on Lexical Acquisition, IJCAI 1989*, ed. U. Zernik. Menlo Park. Morgan Kaufman.

Moens, M., J. Calder, E. Klein, M. Reape, and H. Zeevat. 1989. Expressing Generalizations in Unification-Based Grammar Formalisms. In *Proceedings of the 4th Meeting of the European Chapter of the Association for Computational Linguistics*, 174–181.

Nerbonne, J. 1992. Natural Language Disambiguation and Taxonomic Reasoning. In *DFKI Workshop on Taxonomic Reasoning (DFKI D-92-08)*, ed. J. Heinsohn and B. Hollunder, 40–47. Saarbrücken. DFKI.