Processing Unknown Words in HPSG

Petra Barg and **Markus Walther**[*]
Seminar für Allgemeine Sprachwissenschaft
Heinrich-Heine-Universität Düsseldorf
Universitätsstr. 1, D-40225 Düsseldorf, Germany
{barg,walther}@ling.uni-duesseldorf.de

Abstract

The lexical acquisition system presented in this paper incrementally updates linguistic properties of unknown words inferred from their surrounding context by parsing sentences with an HPSG grammar for German. We employ a gradual, information-based concept of "unknownness" providing a uniform treatment for the range of completely known to maximally unknown lexical entries. "Unknown" information is viewed as revisable information, which is either generalizable or specializable. Updating takes place after parsing, which only requires a modified lexical lookup. Revisable pieces of information are identified by grammar-specified declarations which provide access paths into the parse feature structure. The updating mechanism revises the corresponding places in the lexical feature structures iff the context actually provides new information. For revising generalizable information, type union is required. A worked-out example demonstrates the inferential capacity of our implemented system.

1 Introduction

It is a remarkable fact that humans can often understand sentences containing unknown words, infer their grammatical properties and incrementally refine hypotheses about these words when encountering later instances. In contrast, many current NLP systems still presuppose a complete lexicon. Notable exceptions include Zernik (1989), Erbach (1990), Hastings & Lytinen (1994). See Zernik for an introduction to the general issues involved.

This paper describes an HPSG-based system which can incrementally learn and refine properties of unknown words after parsing individual sentences. It focusses on extracting linguistic properties, as compared to e.g. general concept learning (Hahn, Klenner & Schnattinger 1996). Unlike Erbach (1990), however, it is not confined to simple morpho-syntactic information but can also handle selectional restrictions, semantic types and argument structure. Finally, while statistical approaches like Brent (1991) can gather e.g. valence information from large corpora, we are more interested in full grammatical processing of individual sentences to maximally exploit each context.

The following three goals serve to structure our model. It should i) incorporate a *gradual*, information-based conceptualization of "unknownness". Words are not unknown as a whole, but may contain *unknown, i.e. revisable pieces of information*. Consequently, even known words can undergo revision to e.g. acquire new senses. This view replaces the binary distinction between open and closed class *words*. It should ii) maximally exploit the rich representations and modelling conventions of HPSG and associated formalisms, with essentially the same grammar and lexicon as compared to closed-lexicon approaches. This is important both to facilitate reuse of existing grammars and to enable meaningful feedback for linguistic theorizing. Finally, it should iii) possess domain-independent inference and lexicon-updating capabilities. The grammar writer must be able to fully declare which pieces of information are open to revision.

The system was implemented using MicroCUF, a simplified version of the CUF typed unification formalism (Dörre & Dorna 1993) that we implemented in SICStus Prolog. It shares both the feature logic and the definite clause extensions with its big brother, but substitutes a closed-world type system for CUF's open-world regime. A feature of our type system implementation that will be significant later on is that type information in internal feature struc-

[*]This work was carried out within the *Sonderforschungsbereich 282 'Theorie des Lexikons'* (project B3), funded by the German Federal Research Agency DFG. We thank James Kilbury and members of the B3 group for fruitful discussion.

tures (FSs) can be easily updated.

The HPSG grammar developed with MicroCUF models a fragment of German. Since our focus is on the lexicon, the range of syntactic variation treated is currently limited to simplex sentences with canonical word order. We have incorporated some recent developments of HPSG, esp. the revisions of Pollard & Sag (1994, ch. 9), Manning & Sag (1995)'s proposal for an independent level of argument structure and Bouma (1997)'s use of argument structure to eliminate procedural lexical rules in favour of relational constraints. Our elaborate ontology of semantic types – useful for non-trivial acquisition of selectional restrictions and nominal sorts – was derived from a systematic corpus study of a biological domain (Knodel 1980, 154-188). The grammar also covers all valence classes encountered in the corpus. As for the lexicon format, we currently list full forms only. Clearly, a morphology component would supply more contextual information from known affixes but would still require the processing of unknown stems.

2 Incremental Lexical Acquisition

When compared to a previous instance, a new sentential context can supply either identical, more special, more general, or even conflicting information along a given dimension. Example pairs illustrating the latter three relationships are given under (1)-(3) (words assumed to be unknown in bold face).

(1) a. Im **Axon** tritt ein Ruhepotential auf.
'a rest potential occurs in the **axon**'

b. Das Potential wandert über das **Axon**.
'the potential travels along the **axon**'

(2) a. Das Ohr **reagiert** auf akustische Reize.
'the ear **reacts** to acoustic stimuli'

b. Ein Sinnesorgan **reagiert** auf Reize.
'a sense organ **reacts** to stimuli'

(3) a. Die Nase ist für Gerüche **sensibel**.
'the nose is **sensitive** to smells'

b. Die **sensible** Nase reagiert auf Gerüche.
'the **sensitive** nose reacts to smells'

In contrast to (1a), which provides the information that the gender of *Axon* is not feminine (via *im*), the context in (1b) is more specialized, assigning neuter gender (via *das*). Conversely, (2b) differs from (2a) in providing a more general selectional restriction for the subject of *reagiert*, since sense organs include ears as a subtype. Finally, the adjective *sensibel* is used predicatively in (3a), but attributively in (3b). The usage types must be formally disjoint, because some German adjectives allow for just one usage (*ehemalig* 'former, attr.', *schuld* 'guilty, pred.').

On the basis of contrasts like those in (1)-(3) it makes sense to statically assign revisable information to one of two classes, namely *specializable* or *generalizable*.[1] Apart from the specializable kinds 'semantic type of nouns' and 'gender', the inflectional class of nouns is another candidate (given a morphological component). Generalizable kinds of information include 'selectional restrictions of verbs and adjectives', 'predicative vs attributive usage of adjectives' as well as 'case and form of PP arguments' and 'valence class of verbs'. Note that specializable and generalizable information can cooccur in a given lexical entry. A particular kind of information may also figure in both classes, as e.g. semantic type of nouns and selectional restrictions of verbs are both drawn from the same semantic ontology. Yet the former must be invariantly specialized – independent of the order in which contexts are processed –, whereas selectional restrictions on NP complements should only become more general with further contexts.

2.1 Representation

We require all revisable or updateable information to be expressible as formal types.[2] As relational clauses can be defined to map types to FSs, this is not much of a restriction in practice. Figure 1 shows a relevant fragment. Whereas the combination of specializable information translates into simple type unification (e.g. $non\_fem \wedge neut = neut$), combining

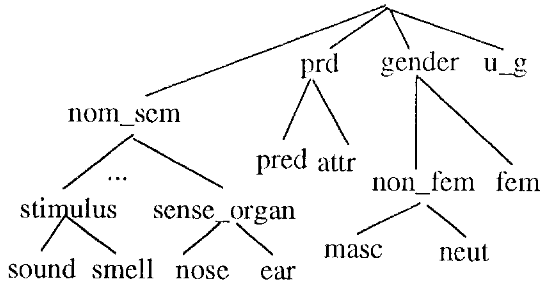

Figure 1: *Excerpt from type hierarchy*

[1] The different behaviour underlying this classification has previously been noted by e.g. Erbach (1990) and Hastings & Lytinen (1994) but received either no implementational status or no systematic association with arbitrary *kinds of information*.

[2] In HPSG types are sometimes also referred to as sorts.

generalizable information requires *type union* (e.g. $pred \vee attr = prd$). The latter might pose problems for type systems requiring the explicit definition of all possible unions, corresponding to least common supertypes. However, type union is easy for (Micro)CUF and similar systems which allow for arbitrary boolean combinations of types. Generalizable information exhibits another peculiarity: we need a disjoint auxiliary type u_g to correctly mark the initial unknown information state.[3] This is because 'content' types like *prd, pred, attr* are to be interpreted as recording what contextual information was encountered in the past. Thus, using any of these to prespecify the initial value – either as the side-effect of a feature appropriateness declaration (e.g. *prd*) or through grammar-controlled specification (e.g. *pred, attr*) – would be wrong (cf. $prd_{initial} \vee attr = prd$, but $u\_g_{initial} \vee attr = u\_g \vee attr$).

Generalizable information evokes another question: can we simply have types like those in fig. 1 within HPSG signs and do in-place type union, just like type unification? The answer is no, for essentially two reasons. First, we still want to rule out ungrammatical constructions through (type) unification failure of coindexed values, so that generalizable types cannot *always* be combined by nonfailing type union (e.g. **der sensible Geruch* 'the sensitive smell' must be ruled out via $sense\_organ \wedge smell = \bot$). We would ideally like to order all type unifications pertaining to a value before all unions, but this violates the order independence of constraint solving. Secondly, we already know that a given informational token can *simultaneously* be generalizable and specializable, e.g. by being coindexed through HPSG's valence principle. However, simultaneous in-place union and unification is contradictory.

To avoid these problems and keep the declarative monotonic setting, we employ two independent features gen and ctxt. ctxt is the repository of contextually unified information, where conflicts result in ungrammaticality. gen holds generalizable information. Since all gen values contain u_g as a type disjunct, they are always unifiable and thus not restrictive during the parse. To nevertheless get correct gen values we perform type union *after* parsing, i.e. during lexicon update. We will see below how this works out.

The last representational issue is how to identify revisable information in (substructures of) the parse FS. For this purpose the grammar defines revisability clauses like the following:

(4) a. generalizable($\boxed{1}$, $\boxed{2}$) :=

$$\left[\text{synsem}\,|\,\text{loc}\,|\,\text{cat}\,|\,\text{head}\begin{bmatrix}\textit{adj}\\ \text{prd}\begin{bmatrix}\text{gen}\ \boxed{1}\\ \text{ctxt}\ \boxed{2}\end{bmatrix}\end{bmatrix}\right]$$

b. specializable($\boxed{1}$) :=

$$\left[\text{synsem}\,|\,\text{loc}\begin{bmatrix}\text{cat}\,|\,\text{head}\ \textit{noun}\\ \text{cont}\,|\,\text{ind}\,|\,\text{gend}\ \boxed{1}\end{bmatrix}\right]$$

2.2 Processing

The first step in processing sentences with unknown or revisable words consists of conventional parsing. Any HPSG-compatible parser may be used, subject to the obvious requirement that lexical lookup must not fail if a word's phonology is unknown. A canonical entry for such unknown words is defined as the disjunction of maximally underspecified generic lexical entries for nouns, adjectives and verbs.

The actual updating of lexical entries consists of four major steps. Step 1 projects the parse FS derived from the whole sentence onto all participating word tokens. This results in word FSs which are contextually enriched (as compared to their original lexicon state) and disambiguated (choosing the compatible disjunct per parse solution if the entry was disjunctive). It then filters the set of word FSs by unification with the right-hand side of revisability clauses like in (4). The output of step 1 is a list of update candidates for those words which were unifiable.

Step 2 determines concrete update values for each word: for each matching *generalizable* clause we take the type union of the gen value of the old, lexical state of the word ($LexGen$) with the ctxt value of its parse projection ($Ctxt$): $TU = LexGen \cup Ctxt$. For each matching *specializable(Spec)* clause we take the parse value *Spec*.

Step 3 checks whether updating would make a difference w.r.t. the original lexical entry of each word. The condition to be met by generalizable information is that $TU \supsetneq LexGen$, for specializable information we similarly require $Spec \subsetneq LexSpec$.

In step 4 the lexical entries of words surviving step 3 are actually modified. We retract the old lexical entry, revise the entry and re-assert it. For words never encountered before, revision must obviously be preceded by making a copy of the generic unknown entry, but with the new word's phonology. Revision itself is the destructive modification of type informa-

[3] Actually, the situation is more symmetrical, as we need a dual type u_s to correctly mark "unknown" *specializable* information. This prevents incorrect updating of known information. However, u_s is unnecessary for the examples presented below.

tion according to the values determined in step 2, at the places in a word FS pointed to by the revisability clauses. This is easy in MicroCUF, as types are implemented via the *attributed variable* mechanism of SICStus Prolog, which allows us to substitute the type in-place. In comparison, general updating of Prolog-encoded FSs would typically require the traversal of large structures and be dangerous if structure-sharing between substituted and unaffected parts existed. Also note that we currently assume DNF-expanded entries, so that updates work on the contextually selected disjunct. This can be motivated by the advantages of working with presolved structures at run-time, avoiding description-level operations and incremental grammar recompilation.

2.3 A Worked-Out Example

We will illustrate how incremental lexical revision works by going through the examples under (5)-(7).

(5) Die **Nase** ist ein Sinnesorgan.
'the **nose** is a sense organ'

(6) Das Ohr **perzipiert**.
'the ear **perceives**'

(7) Eine verschnupfte **Nase perzipiert** den Gestank.
'a bunged up **nose perceives** the stench'

The relevant substructures corresponding to the lexical FSs of the unknown noun and verb involved are depicted in fig. 2. The leading feature paths synsem|loc|cont for *Nase* and synsem|loc|cat|arg-st for *perzipiert* have been omitted.

After parsing (5) the gender of the unknown noun *Nase* is instantiated to *fem* by agreement with the determiner *die*. As the *specializable* clause (4b) matches and the gend parse value differs from its lexical value *gender*, *gender* is updated to *fem*. Furthermore, the object's semantic type has percolated to the subject *Nase*. Since the object's *sense_organ* type differs from generic initial *nom_sem*, *Nase*'s ctxt value is updated as well. In place of the still nonexisting entry for *perzipiert*, we have displayed the relevant part of the generic unknown verb entry.

Having parsed (6) the system then knows that *perzipiert* can be used intransitively with a nominative subject referring to ears. Formally, an HPSG mapping principle was successful in mediating between surface subject and complement lists and the argument list. Argument list instantiations are themselves related to corresponding types by a further mapping. On the basis of this type classification of argument structure patterns, the parse derived the ctxt value *npnom*. Since gen values are *generalizable*, this new value is unioned with the old lexical gen value. Note that ctxt is properly unaffected. The first (subject) element on the args list itself is targeted by another revisability clause. This has the side-effect of further instantiating the underspecified lexical FS. Since selectional restrictions on nominal subjects must become more general with new contextual evidence, the union of *ear* and the old value *u_g* is indeed appropriate.

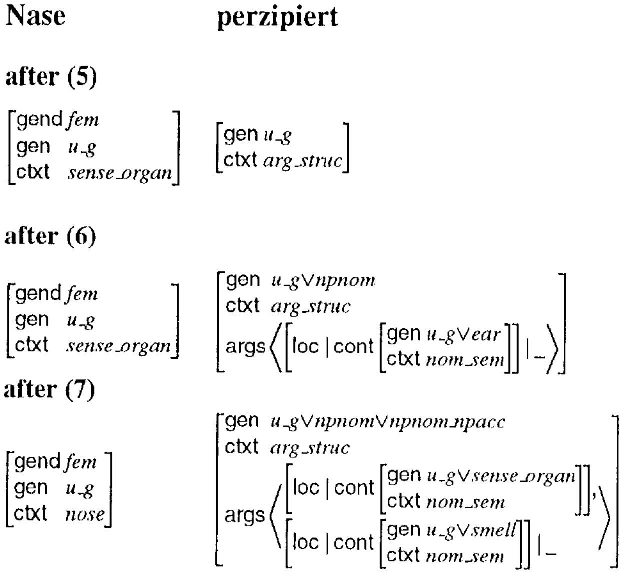

Figure 2: *Updates on lexical FSs*

Sentence (7) first of all provides more specific evidence about the semantic type of partially known *Nase* by way of attributive modification through *verschnupfte*. The system detects this through the difference between lexical ctxt value *sense_organ* and the parse value *nose*, so that the entry is *specialized* accordingly. Since the subject's synsem value is coindexed with the first args element, [ctxt *nose*] simultaneously appears in the FS of *perzipiert*. However, the revisability clause matching there is of class *generalizable*, so union takes place, yielding *ear* ∨ *nose* = *sense_organ* (w.r.t. the simplified ontology of fig. 1 used in this paper). An analogous match with the second element of args identifies the necessary update to be the unioning-in of *smell*, the semantic type of *Gestank*. Finally, the system has learned that an accusative NP object can cooccur with *perzipiert*, so the argument structure type of gen receives another update through union with *npnom_npacc*.

3 Discussion

The incremental lexical acquisition approach described above attains the goals stated earlier. It realizes a gradual, information-based conceptualization of unknownness by providing updateable formal types – classified as either *generalizable* or *specializable* – together with grammar-defined revisability clauses. It maximally exploits standard HPSG representations, requiring moderate rearrangements in grammars at best while keeping with the standard assumptions of typed unification formalisms. One noteworthy demand, however, is the need for a type union operation. Parsing is conventional modulo a modified lexical lookup. The actual lexical revision is done in a domain-independent postprocessing step guided by the revisability clauses.

Of course there are areas requiring further consideration. In contrast to humans, who seem to leap to conclusions based on incomplete evidence, our approach employs a conservative form of generalization, taking the disjunction of actually observed values only. While this has the advantage of not leading to overgeneralization, the requirement of having to encounter all subtypes in order to infer their common supertype is not realistic (sparse-data problem). In (2) *sense_organ* as the semantic type of the first argument of *perzipiert* is only acquired because the simplified hierarchy in fig. 1 has *nose* and *ear* as its only subtypes. Here the work of Li & Abe (1995) who use the MDL principle to generalize over the slots of observed case frames might prove fruitful.

An important question is how to administrate alternative parses and their update hypotheses. In *Das Aktionspotential* ***erreicht*** *den Dendriten* 'the action potential reaches the dendrite(s)', *Dendriten* is ambiguous between acc.sg. and dat.pl., giving rise to two valence hypotheses *npnom_npacc* and *npnom_npdat* for *erreicht*. Details remain to be worked out on how to delay the choice between such alternative hypotheses until further contexts provide enough information.

Another topic concerns the treatment of 'cooccurrence restrictions'. In fig. 2 the system has *independently* generalized over the selectional restrictions for subject and object, yet there are clear cases where this overgenerates (e.g. **Das Ohr perzipiert den Gestank* 'the ear perceives the stench'). An idea worth exploring is to have a partial, extensible list of type cooccurrences, which is traversed by a recursive principle at parse time.

A more general issue is the apparent antagonism between the desire to have both sharp grammatical predictions and continuing openness to contextual revision. If after parsing (7) we transfer the fact that smells are acceptable objects to *perzipiert* into the restricting ctxt feature, a later usage with an object of type *sound* fails. The opposite case concerns newly acquired specializable values. If in a later context these are used to update a gen value, the result may be too general. It is a topic of future research when to consider information certain and when to make revisable information restrictive.

References

Bouma, G. (1997). Valence Alternation without Lexical Rules. In: *Papers from the seventh CLIN Meeting 1996*, Eindhoven, 25–40.

Brent, M. R. (1991). Automatic Acquisition of Subcategorization Frames From Untagged Text. In: *Proceedings of 29th ACL*, Berkeley, 209–214.

Dörre, J. & M. Dorna (1993). CUF – A Formalism for Linguistic Knowledge Representation. In: J. Dörre (Ed.), *Computational Aspects of Constraint-Based Linguistic Description*. IMS, Universität Stuttgart. Deliverable R1.2.A, DYANA-2 – ESPRIT Project 6852.

Erbach, G. (1990). Syntactic Processing of Unknown Words. IWBS Report 131, Institute for Knowledge-Based Systems (IWBS), IBM Stuttgart.

Hahn, U., M. Klenner & K. Schnattinger (1996). Learning from Texts - A Terminological Meta-Reasoning Perspective. In: S. Wermter, E. Riloff & G. Scheler (Ed.), *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, 453–468. Berlin: Springer.

Hastings, P. M. & S. L. Lytinen (1994). The Ups and Downs of Lexical Acquisition. In: *Proceedings of AAAI'94*, 754–759.

Knodel, H. (1980). *Linder Biologie – Lehrbuch für die Oberstufe*. Stuttgart: J.B. Metzlersche Verlagsbuchhandlung.

Li, H. & N. Abe (1995). Generalizing Case Frames Using a Thesaurus and the MDL Principle. In: *Proceedings of Recent Advantages in Natural Language Processing*, Velingrad, Bulgaria, 239–248.

Manning, C. & I. Sag (1995). Dissociations between argument structure and grammatical relations. Ms., Stanford University.

Pollard, C. & I. Sag (1994). *Head-Driven Phrase Structure Grammar*. Chicago University Press.

Zernik, U. (1989). Paradigms in Lexical Acquisition. In: U. Zernik (Ed.), *Proceedings of the First International Lexical Acquisition Workshop*, Detroit.