

# Semantic-Head Based Resolution of Scopal Ambiguities\*

**Björn Gambäck**

Information and Computational Linguistics  
Language Engineering University of Helsinki  
SICS, Box 1263 P.O. Box 4  
S-164 29 Kista, Sweden SF-00014 Helsinki, Finland  
gamback@sics.se

**Johan Bos**

Computational Linguistics  
University of the Saarland  
Postfach 15 11 50  
D-66041 Saarbrücken, Germany  
bos@coli.uni-sb.de

## Abstract

We introduce an algorithm for scope resolution in underspecified semantic representations. Scope preferences are suggested on the basis of semantic argument structure. The major novelty of this approach is that, while maintaining an (scopally) underspecified semantic representation, we at the same time suggest a resolution possibility. The algorithm has been implemented and tested in a large-scale system and fared quite well: 28% of the utterances were ambiguous, 80% of these were correctly interpreted, leaving errors in only 5.7% of the utterance set.

## 1 Introduction

Scopal ambiguities are problematic for language processing systems; resolving them might lead to combinatorial explosion. In applications like transfer-based machine translation, resolution can be avoided if transfer takes place at a representational level encoding scopal ambiguities. The key idea is to have a common representation for *all* the possible interpretations of an ambiguous expression, as in Alshawi et al. (1991). Scopal ambiguities in the source language can then carry over to the target language. Recent research has termed this *underspecification* (see e.g., König and Reyle (1997), Pinkal (1996)).

A problem with underspecification is, however, that structural restrictions are not encoded. Clear scope configurations (preferences) in the source language are easily lost:

- (1) *das paßt auch nicht*  
that fits also not  
'that does not fit either'
- (2) *ich kann<sub>i</sub> sie nicht verstehen*  $\epsilon_i$   
I can you not understand  
'I cannot understand you'

\* This work was funded by BMBF (German Federal Ministry of Education, Science, Research, and Technology) grant 01 IV 101 R. Thanks to Christian Lieske, Scott McGlashan, Yoshiki Mori, Manfred Pinkal, CJ Rupp, and Karsten Worm for many useful discussions.

In (1) the focus particle 'auch' outscopes the negation 'nicht'. The preferred reading in (2) is the one where 'nicht' has scope over the modal 'kann'. In both cases, the syntactic configurational information for German supports the preferred scoping: the operator with the widest scope is c-commanding the operator with narrow scope. Preserving the suggested scope resolution restrictions from the source language would be necessary for a correct interpretation. However, the configurational restrictions do not easily carry over to English; there is no verb movement in the English sentence of (2), so 'not' does not c-command 'can' in this case.

In this paper we focus on the underspecification of scope introduced by quantifying noun phrases, adverbs, and particles. The representations we will use resembles Underspecified Discourse Representation Structures (Reyle, 1993) and Hole Semantics (Bos, 1996).

Our Underspecified Semantic Representation, USR, is introduced in Section 2. Section 3 shows how USRs are built up in a compositional semantics. Section 4 is the main part of the paper. It introduces an algorithm in which structural constraints are used to resolve underspecified scope in USR structures. Section 5 describes an implementation of the algorithm and evaluates how well it fares on real dialogue examples.

## 2 Underspecified Semantics: USR

The representation we will use, USR, is a tertiary term containing the following pieces of semantic information: a top label, a set of labeled conditions, and a set of constraints. The conditions represent ordinary predicates, quantifiers, pronouns, operators, etc., all being uniquely labeled, making it easier to refer to a particular condition. Scope (appearing in quantifiers and operators) is represented in an underspecified way by variables ("holes") ranging over labels.

Labels are written as  $l_n$ , holes as  $h_n$ , and variables over individuals as  $i_n$ . The labelling allows us to state meta-level constraints on the relations between conditions. A constraint  $l \leq h$  is a relation between a label and a hole:  $l$  is either equal to or subordinated to  $h$  (the labeled condition is within the scope denoted by the hole).

$$\langle l_1, \left\{ \begin{array}{l} l_1 : \text{decl}(h_1), \\ l_2 : \text{pron}(i_1), \\ l_3 : \text{passen}(i_2, i_1), \\ l_4 : \text{auch}(h_2), \\ l_5 : \text{nicht}(h_3), \\ l_6 : \text{group}(l_2, l_3) \end{array} \right\}, \left\{ \begin{array}{l} l_4 \leq h_1, \\ l_5 \leq h_1, \\ l_6 \leq h_1, \\ l_6 \leq h_2, \\ l_6 \leq h_3 \end{array} \right\} \rangle$$

(top)                      (conditions)                      (constraints)

Figure 1: The USR for ‘das paßt auch nicht’.

Fig. 1 shows the USR for (1). The top label  $l_1$  introduces the entire structure and points to the declarative sentence mood operator, outscoping all other elements. The pronoun ‘das’ is *pron*, marking unresolved anaphora. ‘auch’ and ‘nicht’ are handled as operators. The verb condition (*passen*) and its pronoun subject are in the same scope unit, represented by a grouping.

The first three constraints state that neither the verb, nor the two particles outscope the mood operator. The last two put the verb information in the scope of the particles. (NB: no restrictions are placed on the particles’ relative scope.) Fig. 2 shows the subordination relations.

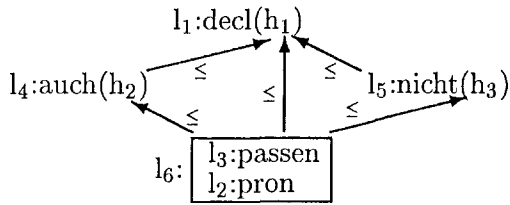


Figure 2: Scopal relations in the USR.

A USR is interpreted with respect to a “*plugging*”, a mapping from holes to labels (Bos, 1996). The number of readings the USR encodes equals the number of possible pluggings. Here, two pluggings do not violate the  $\leq$  constraints:

$$\begin{array}{l} (3) \quad \{h_1 = l_4, \quad h_2 = l_5, \quad h_3 = l_6\} \\ (4) \quad \{h_1 = l_5, \quad h_2 = l_6, \quad h_3 = l_4\} \end{array}$$

The plugging in (3) resembles the reading where ‘auch’ outscopes ‘nicht’: the label for ‘nicht’,  $l_5$ , is taken to “plug” the hole for ‘auch’,  $h_2$ , while ‘auch’ ( $l_4$ ) is plugging the top hole of the sentence,  $h_1$ . In contrast, the plugging in (4) gives the reading where the negation has wide scope.

With a plugging, a USR can be translated to a Discourse Representation Structure, DRS (Kamp and Reyle, 1993): a *pron* condition introduces a discourse marker which should be linked to an antecedent, *group* is a *merge* between DRSs, *passen* a one place predicate, etc.

### 3 Construction of USRs

In addition to underspecification, we let two other principles guide the semantic construction: *lexicalization* (keep as much as possible of the semantics lexicalized) and *compositionality* (a phrase’s interpretation is a function of its sub-phrases’ interpretations). The grammar rules allow for addition of already manifest information (e.g., from the lexicon) and three ways of passing non-manifest information (e.g., about complements sought): trivial composition, functor-argument and modifier-argument application.

*Trivial composition* occurs in grammar rules which are semantically unary branching, i.e., the semantics of at the most one of the daughter (right-hand side) nodes need to influence the interpretation of the mother (left-hand side) node.

The application type rules appear on semantically binary branching rules: In *functor-argument application* the bulk of the semantic information is passed between the mother node and the functor (semantic head). In *modifier-argument application* the argument is the semantic head, so most information is passed up from that. (Most notably, the label identifying the entire structure will be the one of the head daughter. We will refer to it as the *main label*.)

The difference between the two application types pertains to the (semantic) subcategorization schemes: In functor-argument application (5), the functor subcategorizes for the argument, the argument may optionally subcategorize for the functor, and the mother’s subcategorization list is the functor’s, minus the argument:

$$(5) \quad \begin{array}{c} \text{Mother} \\ \left[ \begin{array}{l} \text{main-label} \quad \boxed{1} \\ \text{subcat} \quad \langle \boxed{3} \rangle \end{array} \right] \Rightarrow \\ \begin{array}{c} \text{Functor (head)} \\ \left[ \begin{array}{l} \text{main-label} \quad \boxed{1} \\ \text{subcat} \quad \langle \boxed{2} \mid \boxed{3} \rangle \end{array} \right] \\ \text{Argument (nonhead)} \\ \left[ \begin{array}{l} \text{main-label} \quad \boxed{2} \\ \text{subcat} \quad \langle \boxed{1} \rangle \vee \langle \quad \rangle \end{array} \right] \end{array} \end{array}$$

In modifier-argument application (6), *Modifier* subcategorizes for *Argument* (only), while *Argument* does *not* subcategorize for *Modifier*. Its *subcat* list is passed unchanged to *Mother*.

$$(6) \left[ \begin{array}{c} \text{Mother} \\ \text{main-label } \boxed{1} \\ \text{subcat } \boxed{2} \end{array} \right] \Rightarrow \left[ \begin{array}{c} \text{Modifier (nonhead)} \\ \text{main-label } \textit{Label} \\ \text{subcat } \boxed{1} \end{array} \right] \left[ \begin{array}{c} \text{Argument (head)} \\ \text{main-label } \boxed{1} \\ \text{subcat } \boxed{2} \end{array} \right]$$

## 4 A Resolution Algorithm

Previous approaches to scopal resolution have mainly been treating the scopal constraints separately from the rest of the semantic structure and argued that contextual information must be taken into account for correct resolution. However, the SRI Core Language Engine used a straight-forward approach (Moran and Pereira, 1992). Variables for the unresolved scopal constraints were asserted at the lexical level together with some constraints on the resolution. Constraints could also be added in grammar rules, albeit in a somewhat *ad hoc* manner. Most of the scopal resolution constraints were, though, provided by a separate knowledge-base specifying the inter-relation of different scope-bearing operators. The constraints were applied in a process subsequent to the semantic construction.

### 4.1 Lexical entries

In contrast, we want to be able to capture the constraints already given by the function-argument structure of an utterance and provide a possible resolution of the scopal ambiguities. This resolution should be built up during the construction of (the rest of) the semantic representation. Thus we introduce a set of features (called **holeinfo**) on each grammatical category.

On terminals, the features in this set will normally have the values shown in (7), indicating that the category does not contain a hole (**isa-hole** has the value *no*), i.e., it is a nonscope-bearing element. **sb-label**, the *semantic-head based resolution label*, is the label of the element of the substructure below it having widest scope. In the lexicon, it is the entry's own main label.

$$(7) \left[ \text{holeinfo} \left[ \begin{array}{c} \text{sb-label } \textit{MainLabel} \\ \text{isa-hole } \textit{no} \\ \text{hole } \textit{no} \end{array} \right] \right]$$

Scope-bearing categories (quantifiers, particles, etc.) introduce holes and get the feature setting of (8). The feature **hole** points to the hole introduced. (Finite verbs are also treated this way: they are assumed to introduce a hole for the scope of the sentence mood operator.)

$$(8) \left[ \text{holeinfo} \left[ \begin{array}{c} \text{sb-label } \textit{MainLabel} \\ \text{isa-hole } \textit{yes} \\ \text{hole } \textit{Hole} \end{array} \right] \right]$$

### 4.2 Grammar rules

When the **holeinfo** information is built up in the analysis tree, the **sb-labels** are passed up as the main labels (i.e., from the semantic head daughter to the mother node), unless the nonhead daughter of a binary branching node contains a hole. In that case, the hole is plugged with the **sb-label** of the head daughter and the **sb-label** of the mother node is that of the nonhead daughter. The effect being that a scope-bearing nonhead daughter is given scope over the head daughter. On the top-most level of the grammar, the hole of the sentence mood operator is plugged with the **sb-label** of the full structure.

Concretely, grammar rules of both application types pass **holeinfo** as follows. If the nonhead daughter does not contain a hole, **holeinfo** is unchanged from head daughter to mother node:

$$(9) \left[ \begin{array}{c} \text{Mother} \\ \text{holeinfo } \boxed{1} \end{array} \right] \Rightarrow \left[ \begin{array}{c} \text{Head} \\ \text{holeinfo } \boxed{1} \end{array} \right] \left[ \begin{array}{c} \text{Nonhead} \\ \text{holeinfo } [\text{isa-hole } \textit{no}] \end{array} \right]$$

However, if the nonhead daughter *does* contain a hole, it is plugged with the **sb-label** of the head daughter and the mother node gets its **sb-label** from the nonhead daughter. The rest of the **holeinfo** still come from the head daughter:

$$(10) \left[ \begin{array}{c} \text{Mother} \\ \text{sb-label } \boxed{1} \\ \text{isa-hole } \boxed{2} \\ \text{hole } \boxed{3} \end{array} \right] \Rightarrow \left[ \begin{array}{c} \text{Head} \\ \text{sb-label } \textit{HeadLabel} \\ \text{isa-hole } \boxed{2} \\ \text{hole } \boxed{3} \end{array} \right] \left[ \begin{array}{c} \text{Nonhead} \\ \text{sb-label } \boxed{1} \\ \text{isa-hole } \textit{yes} \\ \text{hole } \textit{Hole} \end{array} \right]$$

The hole to be plugged is here identified by the **hole** feature of the nonhead daughter. To show the preferred scopal resolution, a relation ' $\text{Hole} =_{sb} \text{HeadLabel}$ ', a *semantic-head based plugging*, is introduced into the USR.

### 4.3 Resolution Example

We will illustrate the rules with an example. The utterance (1) '*das paßt auch nicht*' has the semantic argument structure shown in Fig. 3, where  $\text{Node}[L, H]$  stands for the node  $\text{Node}$  having an **sb-label**  $L$  and **hole** feature value  $H$ .

The verb *passen* is first applied to the subject '*das*'. The **sb-label** of '*passen*' is its main label

(the grouping label  $l_6$ ). Its **hole** feature points to  $h_1$ , the mood operator’s scope unit. The pronoun contains no hole (is nonscope-bearing), so we have the first case above, rule (9), in which the mother node’s **holeinfo** is identical to that of the head daughter, as indicated in the figure.

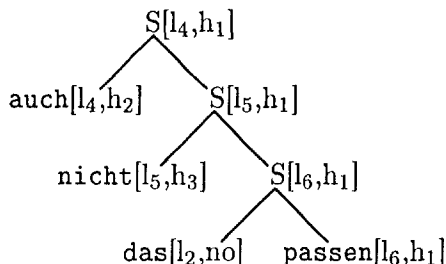


Figure 3: Semantic argument structure

Next, the modifier ‘*nicht*’ is applied to the verbal structure, giving the case with the nonhead daughter containing a hole, rule (10). For this hole we add a ‘ $h_3 =_{sb} l_6$ ’ to the USR: The label plugging the hole is the **sb-label** of the head daughter. The **sb-label** of the resulting structure is  $l_5$ , the **sb-label** of the modifier. The process is repeated for ‘*auch*’ so that its hole,  $h_2$ , is plugged with  $l_5$ , the label of its argument. We have reached the end of the analysis and  $h_1$ , the remaining hole of the entire structure is plugged by the structure’s **sb-label**, which is now  $l_4$ . In total, three semantic-head based plugging constraints are added to the USR in Fig. 1:

$$(11) \quad h_1 =_{sb} l_4, \quad h_2 =_{sb} l_5, \quad h_3 =_{sb} l_6$$

Giving a scope preference corresponding to the plugging (3), the reading with *auch* outscoping *nicht*, resulting in the correct interpretation.

#### 4.4 Coordination

Sentence coordinations, discourse relation adverbs, and the like add a special case. These categories force the scopal elements of their sentential complements to be resolved locally, or in other words, introduce a new hole which should be above the top holes of both complements. They get the lexical setting

$$(12) \quad \left[ \text{holeinfo} \left[ \begin{array}{ll} \text{sb-label} & \text{MainLabel} \\ \text{isa-hole} & \text{island} \\ \text{hole} & \text{Hole} \end{array} \right] \right]$$

So, **isa-hole** indicates which type of hole a structure contains. The values are **no**, **yes**, and **island**. **island** is used to override the argument structure to produce a plugging where

the top holes of the sentential complements get plugged with their own **sb-labels**. This complicates the implementation of rules (9) and (10) a bit; they must also account for the fact that a daughter node may carry an **island** type hole.

## 5 Implementation and Evaluation

The resolution algorithm described in Section 4 has been implemented in Verbmobil, a system which translates spoken German and Japanese into English (Bub et al., 1997). The underspecified semantic representation technique we have used in this paper reflects the core semantic part of the Verbmobil Interface Term, VIT (Bos et al., 1998). The aim of VIT is to describe a consistent interface structure between the different language analysis modules within Verbmobil. Thus, in contrast to our USR, VIT is a representation that encodes *all* the linguistic information of an utterance; in addition to the USR semantic structure of Section 2, the Verbmobil Interface Term contains prosodic, syntactic, and discourse related information.

In order to evaluate the algorithm, the results of the pluggings obtained for four dialogues in the Verbmobil test set were checked (Table 1).

We only consider utterances for which the VITs contain more than two holes: The number of scope-bearing operators is the number of holes minus one. Thus, a VIT with one hole only trivially contains the top hole of the utterance (i.e., the hole for the sentence mood predicate; introduced by the main verb).

A VIT with two holes contains the top hole and the hole for one scope-taking element. However, the mood-predicate will always have scope over the remaining proposition, so resolution is still trivial.

Table 1: Results of evaluation

Dial. Id.	# Utt.	# Correct	utt. / # holes	%		
		$\leq 2$	3	$\geq 5$		
B1	48	34	9/11	1/2	1/1	79
B2	41	26	5/8	2/3	4/4	73
B7	48	36	7/8	0/1	3/3	83
RHQ1	91	68	10/11	5/6	4/6	83
Total	228	164	31/38	8/12	12/14	80

The dialogues evaluated are identified as three of the ‘Blaubeuren’ dialogues (B1, B2, and B7) and one of the ‘Reithinger-Herweg-Quantz’ dialogues (RHQ1). These four together form the standard test-set for the German language modules of the Verbmobil system.

For VITs with three or more holes, we have true ambiguities. Column 3 gives the number of utterances with no ambiguity ( $\leq 2$  holes), the columns following look at the ambiguous sentences. Most commonly the utterances contained one true ambiguity (3 holes, as in Fig. 2). Utterances with more than two ambiguities ( $\geq 5$  holes) are rare and have been grouped together.

Even though the algorithm is fairly straightforward, resolution based on semantic argument structure fares quite well. Only 64 (28%) of the 228 utterances are truly ambiguous (i.e., contain more than two holes). The default scoping introduced by the algorithm is the preferred one for 80% of the ambiguous utterances, leaving errors in just 13 (5.7%) of the utterances overall.

Looking closer at these cases, the reasons for the failures divide as: the relative scope of two particles did not conform to the c-command structure assigned by syntax (one case); an indefinite noun phrase should have received wide scope (3), or narrow scope (1); an adverb should have had wide scope (3); combination of (a modal) verb movement and negated question (1); technical construction problem in VIT (4).

The resolution algorithm has been implemented in Verbmobil in both the German semantic processing (Bos et al., 1996) and the (substantially smaller) Japanese one (Gambäck et al., 1996). Evaluating the performance of the resolution algorithm on the standard test suite for the Japanese parts of Verbmobil (the “RDS1” reference dialogue), we found that only 7 of the 36 sentences in the dialogue contained more than two holes. All but one of the ambiguities were correctly resolved by the algorithm. Even though the number of sentences tested certainly is too small to draw any real conclusions from, the correctness rate still indicates that the algorithm is applicable also to Japanese.

## 6 Conclusions

We have presented an algorithm for scope resolution in underspecified semantic representations. Scope preferences are suggested on the basis of semantic argument structure, letting the nonhead daughter node outscope the head daughter in case both daughter nodes are scope-bearing. The algorithm was evaluated on four “real-life” dialogues and fared quite well: about 80% of the utterances containing scopal ambi-

guities were correctly interpreted by the suggested resolution, leaving scopal resolution errors in only 5.7% of the overall utterances.

The algorithm is computationally cheap and quite straight-forward, yet its predictions are relatively accurate. Our results indicate that for a practical system, more sophisticated approaches to scopal resolution (i.e., based on the relations between different scope-bearing elements and/or contextual information) will not add much to the overall system performance.

## References

- Alshawi H., D.M. Carter, B. Gambäck, and M. Rayner. 1991. Translation by quasi logical form transfer. *Proc. 29th ACL*, pp. 161–168, University of California, Berkeley.
- Bos J. 1996. Predicate logic unplugged. *Proc. 10th Amsterdam Colloquium*, pp. 133–142, University of Amsterdam, Holland.
- Bos J., B. Gambäck, C. Lieske, Y. Mori, M. Pinkal, and K. Worm. 1996. Compositional semantics in Verbmobil. *Proc. 16th COLING*, vol. 1, pp. 131–136, København, Denmark.
- Bos J., B. Buschbeck-Wolf, M. Dorna, and C.J. Rupp. 1998. Managing information at linguistic interfaces. *Proc. 17th COLING and 36th ACL*, Montreal, Canada.
- Bub T., W. Wahlster, and A. Waibel. 1997. Verbmobil: The combination of deep and shallow processing for spontaneous speech translation. *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 71–74, München, Germany.
- Gambäck B., C. Lieske, and Y. Mori. 1996. Underspecified Japanese semantics in a machine translation system. *Proc. 11th Pacific Asia Conf. on Language, Information and Computation*, pp. 53–62, Seoul, Korea.
- Kamp H. and U. Reyle. 1993. *From Discourse to Logic*. Kluwer, Dordrecht, Holland.
- König E. and U. Reyle. 1997. A general reasoning scheme for underspecified representations. In H. J. Ohlbach and U. Reyle, eds, *Logic and its Applications. Festschrift for Dov Gabbay. Part I*. Kluwer, Dordrecht, Holland.
- Moran D.B. and F.C.N. Pereira. 1992. Quantifier scoping. In Alshawi H., ed. *The Core Language Engine*. The MIT Press, Cambridge, Massachusetts, pp. 149–172.
- Pinkal M. 1996. Radical underspecification. *Proc. 10th Amsterdam Colloquium*, pp. 587–606, University of Amsterdam, Holland.
- Reyle U. 1993. Dealing with ambiguities by underspecification: Construction, representation and deduction. *Journal of Semantics*, 10:123–179.