

Ambiguity Preserving Machine Translation using Packed Representations*

Martin C. Emele and Michael Dorna

IMS, Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Azenbergstraße 12
D-70174 Stuttgart
{emele,dorna}@ims.uni-stuttgart.de

Abstract

In this paper we present an ambiguity preserving translation approach which transfers ambiguous LFG f-structure representations. It is based on packed f-structure representations which are the result of potentially ambiguous utterances. If the ambiguities between source and target language can be preserved, no unpacking during transfer is necessary and the generator may produce utterances which maximally cover the underlying ambiguities. We convert the packed f-structure descriptions into a flat set of prolog terms which consist of predicates, their predicate argument structure and additional attribute-value information. Ambiguity is expressed via local disjunctions. The flat representations facilitate the application of a Shake-and-Bake like transfer approach extended to deal with packed ambiguities.

1 Introduction

It is a central problem for any practical NLP system and specifically for any machine translation (MT) system to deal with ambiguity of natural language utterances. This is especially true for systems with large coverage grammars, where the number of potentially ambiguous descriptions grows dramatically as the number of acceptable syntactic constructions and the number of lexical readings increases. In general, it is not possible to resolve all potentially ambiguous descriptions without incorporating world knowledge of unlimited size. This fundamental problem has been discussed in the litera-

ture as the AI completeness problem (cf. Kay et al. (1994)). Nevertheless, it has been observed that many ambiguous utterances in the source language (SL) text can be translated by equivalently ambiguous phrases in the target language (TL) text. We call such an ambiguity a *preservable ambiguity* and the corresponding architecture for translation an ambiguity preserving MT approach.

In order to achieve this goal of ambiguity preserving translations there exist a number of different solutions we can apply. A naive solution would enumerate all possible ambiguous descriptions, translate them and generate the corresponding target utterances which would then be intersected to find a common string which covers all meanings. This strategy is obviously not feasible because the number of potential readings might grow exponentially with the length of the sentence.

Another solution to overcome this problem is not to resolve ambiguities at all by using underspecified representations. This strategy has been successfully applied for a number of semantic ambiguities like quantifier and operator scope ambiguities. Therefore it is not surprising that the usage of underspecified semantic representations have gained much popularity in recent years. Work in the literature include the QLF representations (Alshawi, 1992), the work on Underspecified Discourse Representation Structures (UDRS) (Reyle, 1993; Bos et al., 1996), and the collection of papers in van Deemter and Peters (1996). For an application of using underspecified semantic representations within MT see Alshawi et al. (1991), Copestake et al. (1995) and Dorna and Emele (1996).

Another source of ambiguities which might be preservable between related languages include syntactic ambiguities like the well-known PP at-

* We would like to thank our colleagues at Xerox PARC and Xerox RCE for fruitful discussions and the anonymous reviewers for valuable feedback. This work was funded by the German Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the Verbmobil project under grant 01 IV 701 N3.

tachment ambiguities. There has been growing interest in developing underspecified or so called *packed* representations to deal with such syntactic ambiguities (cf. Rich et al. (1987), Seo and Simmons (1989), Bear and Hobbs (1988), Maxwell III and Kaplan (1993), Pinkal (1995), Egg and Lebeth (1995), Schiehlen (1996) and Dörre (1997)).

The key idea of all these representations is to factor common information as much as possible in a parse forest and to represent the attachment ambiguities as local disjunctions without conversion to disjunctive normal form. Such representations avoid the exponential explosion which would result if all possible readings are extracted from the parse forest.

To achieve our overall goal of ambiguity preserving MT it requires not only a parser which is able to produce such packed representations but also a generator which is able to take such a packed representation as input and generate all possible paraphrases without explicitly enumerating all readings. The work in Kay (1996) and the extension to ambiguous input in Shemtov (1996) and Shemtov (1997) describes a chart-based generation process which takes packed representations as input and generates all paraphrases without expanding first into disjunctive normal form.

What needs to be done to realize our envisaged goal is a transfer system which is able to work on these packed translations without unpacking them or only as much as necessary if ambiguities can only partly be preserved in the target language. The rest of this paper is concerned with the extension of a Shake-and-Bake like transfer approach (Whitelock, 1992; Beaven, 1992) or the kind of semantic-based transfer approach as described for example in Dorna and Emele (1996) to cope with local ambiguities.

To explain and illustrate the treatment of local ambiguities we show how an underspecified representation of PP attachment ambiguities can be utilized in a machine translation architecture for providing ambiguity preserving translations. It is illustrated on the basis of LFG f-structure level representations (Kaplan and Bresnan, 1982). However, it could equally well be done on the level of underspecified semantic representations as shown in (Dorna et

al., 1998). The main reason for choosing the f-structure level representation is due to the fact that we could use the Xerox Linguistic Environment (XLE) system (Maxwell III and Kaplan, 1996) for the analysis and generation of English and German utterances. The key argument for using this linguistic workbench is the ability to produce packed representations for ambiguous utterances using techniques described in Maxwell III and Kaplan (1993) and the availability of a generator which generates utterances from f-structure descriptions.

The rest of the paper is structured as follows: first, we show how the hierarchical f-structure representations can be converted into a flat set of Prolog predicates such that the Shake-and-Bake like transfer approach can be applied. Second, we show how PP attachment ambiguities are represented using a packed representation. Then we show how this particular transfer approach can be adopted for dealing with this kind of ambiguous representations.

2 Example

To illustrate the approach we take a simple example which contains a PP attachment ambiguity which can be preserved between German and English and probably between many other related languages as well.

- (1) wir treffen die Kollegen in Berlin
we meet the colleagues in Berlin

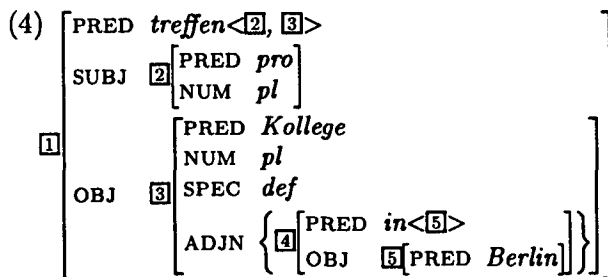
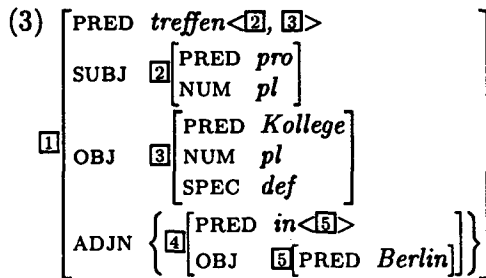
For example the sentence in (1) can either mean (a) that we will have a meeting in Berlin where we will meet our colleagues or (b) that we will meet our colleagues who live in Berlin. Without previous knowledge about the discourse and the specific people involved, it will not be possible to resolve these two meanings. Nevertheless, both the German and the English sentence express exactly the same ambiguity.

There might exist other paraphrases using exactly the same semantic predicates, e.g. the utterances in (2) but they will not be chosen by the generator because they do not cover both readings at the same time. Instead sentence (2a) would be chosen to express the attachment of the prepositional phrase to the verb phrase whereas sentence (2b) would be chosen to express the attachment to the noun phrase 'the colleagues'.

- (2) a. In Berlin treffen wir die Kollegen
 In Berlin meet we the colleagues
 (In Berlin we will meet the colleagues.)
- b. wir treffen die Kollegen aus Berlin
 we meet the colleagues from Berlin
 (We will meet the colleagues from Berlin.)

In addition, those two maximally discriminating sentences could also be used as an interface for an interactive translation system, e.g. the negotiator approach (Kay, 1997) where the human translator would be asked to distinguish between the two possible readings.

The f-structures in (3) and (4) correspond to the disambiguated attachments as paraphrased in (2a) and (2b) respectively.



3 From F-structures to Term Sets

F-structures encode information in a hierarchical manner by recursively embedding substructures. They provide by nature only outside-in references whereas in transfer frequently inside-out access is necessary. Hence, information access for transformation processes like transfer is not as straightforward as it could be when using flat set representations (Beaven, 1992; Whitelock, 1992). Set representations can be seen as a pool of constraints where co-references between the constraints, i.e. the set elements, are used to encode the same embedding f-structures provide. Therefore, the structural embedding which is, on the one hand, part of f-structures themselves is represented, on the other hand, in the interpretation of constraint sets. Furthermore, sets come with very simple test and manipulation

operations such as tests for membership and set union.

In the following we define a correspondence between f-structures and sets of terms. We restrict the f-structures to transfer relevant information such as PREDs, grammatical functions, etc. Feature structure constraints are encoded as relational constraints using Prolog syntax (cf. Johnson (1991)). As examples of such sets of terms see (5) and (6) which corresponds to f-structures (3) and (4), respectively.

- (5) `treffen(1),
 subj(1,2),pro(2),num(2,pl),
 obj(1,3),kollege(3),num(3,pl),
 spec(3,def),
 adjn(1,4),in(4),
 obj(4,5),Berlin(5)`
- (6) `treffen(1),
 subj(1,2),pro(2),num(2,pl),
 obj(1,3),kollege(3),num(3,pl),
 spec(3,def),
 adjn(3,4),in(4),
 obj(4,5),Berlin(5)`

The 2-place relation *trans* given below translates between f-structures and (sets of) terms. \boxed{i} are references to f-structures which are mapped into nodes i used in terms. Γ are features, $\Pi(\dots)$ describe predicates, v stands for atomic values, and φ are complex f-structures. Co-occurring parts of f-structures are translated only once.

1. (atomic values)
 $\text{trans} \langle \boxed{i}[\Gamma v], \Gamma(i, v) \rangle$
2. (predicate values)
 $\text{trans} \langle \boxed{i}[\text{PRED } \Pi(\dots)], \Pi(i) \rangle$
3. (complex f-structure values)
 $\text{trans} \langle \boxed{i}[\Gamma \boxed{j} \varphi], \Gamma(i, j) \cup T \rangle$
 with $\text{trans} \langle \boxed{j} \varphi, T \rangle$
4. (set values)
 $\text{trans} \langle \boxed{i}[\text{ADJN } \{ \boxed{i_1} \varphi_1, \dots, \boxed{i_n} \varphi_n \}],$
 $\text{adjn}(i, i_1), \dots, \text{adjn}(i, i_n)$
 $\cup T_1 \cup \dots \cup T_n \rangle$
 with $\text{trans} \langle \boxed{i_j} \varphi_j, T_j \rangle; 1 \leq j \leq n$

trans is bidirectional, i.e. we are able to translate between f-structures and terms for using terms as transfer input, process terms in the transfer, and convert the transfer output back to f-structures which are the appropriate generator representations.

4 F-structure Transfer

Transfer works on source language (SL) and target language (TL) sets of terms representing predicates, roles, etc. like the ones shown in (5) and (6). The mapping is encoded in transfer rules as in (7). For a rule to be applied, the set on the SL side must be a matching subset of the SL input set. If this is the case, we remove the covering set from the input and add the set on the other side of the rule to the TL output. Transfer is complete, if the SL set is empty.

- (7) a. $\text{treffen}(E) \leftrightarrow \text{meet}(E)$.
 b. $\text{kollege}(X) \leftrightarrow \text{colleague}(X)$.
 c. $\text{Berlin}(X) \leftrightarrow \text{Berlin}(X)$.
 d. $\text{in}(X) \leftrightarrow \text{in}(X)$.
 e. $\text{pro}(X) \leftrightarrow \text{pro}(X)$.
 f. $\text{subj}(X,Y) \leftrightarrow \text{subj}(X,Y)$.
 g. $\text{obj}(X,Y) \leftrightarrow \text{obj}(X,Y)$.
 h. $\text{adjn}(X,Y) \leftrightarrow \text{adjn}(X,Y)$.

The transfer operator \leftrightarrow is bidirectional. Upper case letters in argument positions are logical variables which will be bound to nodes at runtime. Because of the variable sharings on both sides of a rule we work on the same nodes of a graph. Hence, the overall mechanism can be formalized as a graph rewriting process.

- (8) a. $\text{meet}(1)$,
 $\text{subj}(1,2), \text{pro}(2), \text{num}(2, \text{pl})$
 $\text{obj}(1,3), \text{colleague}(3)$,
 $\text{num}(3, \text{pl}), \text{spec}(3, \text{def})$,
 $\text{adjn}(1,4), \text{in}(4)$
 $\text{obj}(4,5), \text{Berlin}(5)$

- b.
$$\left[\begin{array}{l} \text{PRED } \textit{meet} \langle [2], [3] \rangle \\ \text{SUBJ } [2] \left[\begin{array}{l} \text{PRED } \textit{pro} \\ \text{NUM } \textit{pl} \end{array} \right] \\ \text{OBJ } [3] \left[\begin{array}{l} \text{PRED } \textit{colleague} \\ \text{NUM } \textit{pl} \\ \text{SPEC } \textit{def} \end{array} \right] \\ \text{ADJN } \left\{ [4] \left[\begin{array}{l} \text{PRED } \textit{in} \langle [5] \rangle \\ \text{OBJ } [5] \left[\text{PRED } \textit{Berlin} \end{array} \right] \end{array} \right\} \right\}$$

Applying the rule set in (7) to (5), we yield the result in (8a). Using the correspondence between f-structures and term representations it is possible to translate back to the TL f-structure in (8b). This f-structure will be passed on to the generator which will produce the utterance in (2a) as one of the possible paraphrases.

The transfer rules in (7c-h) which are defined as the identity transformation between SL and

TL are actually redundant. They can be replaced via a general metarule which passes on all singleton sets which are not covered by any explicit transfer rule. The same metarule transfers also morpho-syntactic information like number and definiteness.

5 Packed Representations

The following example in (9) provides a packed f-structure representation for the German sentence in (1). The ambiguous PP attachment of the 'in' PP is represented via a local disjunction¹ ($X=1 \vee X=3$) which binds the external variable X of the adjunct relation to either node 1 or node 3 representing the VP or NP attachment, respectively.

- (9) a. $\text{treffen}(1)$,
 $\text{subj}(1,2), \text{pro}(2), \text{num}(2, \text{pl})$
 $\text{obj}(1,3), \text{kollege}(3)$,
 $\text{num}(3, \text{pl}), \text{spec}(3, \text{def})$,
 $\text{adjn}(X,4), \text{in}(4)$
 $\text{obj}(4,5), \text{Berlin}(5)$,
 $(X=1 \vee X=3)$

- b.
$$\left[\begin{array}{l} \text{PRED } \textit{treffen} \langle [2], [3] \rangle \\ \text{SUBJ } [2] \left[\begin{array}{l} \text{PRED } \textit{pro} \\ \text{NUM } \textit{pl} \end{array} \right] \\ \text{OBJ } [3] \left[\begin{array}{l} \text{PRED } \textit{Kollege} \\ \text{NUM } \textit{pl} \\ \text{SPEC } \textit{def} \end{array} \right] \\ \text{ADJN } \left\{ [4] \left[\begin{array}{l} \text{PRED } \textit{in} \langle [5] \rangle \\ \text{OBJ } [5] \left[\text{PRED } \textit{Berlin} \end{array} \right] \end{array} \right\} \right\} \\ \text{[X]=[1] } \vee \text{ [X]=[3]} \end{array} \right]$$

Applying the very same transfer rules in (7) to the input in (9) produces the result in (10) which fully preserves the ambiguity between source and target language.

- (10) $\text{meet}(1)$,
 $\text{subj}(1,2), \text{pro}(2), \text{num}(2, \text{pl})$
 $\text{obj}(1,3), \text{colleague}(3)$,
 $\text{num}(3, \text{pl}), \text{spec}(3, \text{def})$,
 $\text{adjn}(X,4), \text{in}(4)$
 $\text{obj}(4,5), \text{Berlin}(5)$,
 $(X=1 \vee X=3)$

If the generator takes the corresponding f-structure for this packed description as input it will generate (1) repeated in (11) and not any of

¹The notation of using a local disjunction is used only for illustrating purposes. The actual implementation uses contexted constraints as developed and implemented in the XLE system (cf. Maxwell III and Kaplan (1991)).

the paraphrases in (2) because they would not cover both ambiguities at the same time.

(11) We will meet the colleagues in Berlin.

The local disjunction is not affected by the application of the transfer rule for mapping the adjunct relation to the target language because there is no interaction between the variable X and any other predicate.

6 Local Disambiguation

If it is not possible to fully preserve the attachment ambiguities between source and target language, we need to partially disambiguate the relevant ambiguity. For example, this would be the case if we would translate (1) to Japanese. Depending whether we attach to the NP 'the colleagues' or to the VP we have to choose between two different postpositions 'de' (location) vs. 'no' (adnominal modification). The two sentences in (12) show the Japanese translations together with their English glosses.

- (12) a. watashi tachi -ga berurin -de
 we NOM Berlin LOC
 dooryoo -to aimasu
 colleagues COM will meet
 (In Berlin we will meet the colleagues.)
- b. watashi tachi -ga berurin -no
 we NOM Berlin MOD
 dooryoo -to aimasu
 colleagues COM will meet
 (We will meet the colleagues from Berlin.)

The choice of the postposition could be triggered via selectional restrictions in the condition part of the transfer rules. The rules in (13) show two components on their lefthand sides: the part to the right of # is a test on a copy of the original input. The test matches an adjunct relation where the variable Y is bound to the internal argument. Y is coindexed with the node of the SL preposition 'in'. The variable X is bound to the external argument node where the adjunct is attached. The second element of the test checks the selectional restriction² of this attachment.

²Instead of using explicit predicates for testing selectional restrictions the real system uses a sort system. The test on explicit predicates is replaced with a more general sortal subsumption test, e.g. $sort(X) \leq_{event}$ vs. $sort(X) \leq_{object}$.

- (13) a. $in(Y) \# adjn(X,Y), treffen(X) \rightarrow de(Y)$.
 b. $in(Y) \# adjn(X,Y), kollege(X) \rightarrow no(Y)$.

The Japanese distinction is parallel to the case where the German preposition 'in' would be translated either with the English preposition 'in' or the preposition 'from' depending which of the two meanings is taken. Hence for ease of exposition we will apply the two equivalent transfer rules in (14) for the translation of the 'in' instead of the equivalent Japanese ones.

- (14) a. $in(Y) \# adjn(X,Y), treffen(X) \rightarrow in(Y)$.
 b. $in(Y) \# adjn(X,Y), kollege(X) \rightarrow from(Y)$.

Since the external argument of the adjunct relation takes part in the local disjunction ($X=1 \vee X=3$) the application of transfer rule (14a) triggers a local resolution. This is done by applying the distributive law such that the selectional restriction can be tested. For the first disjunct this yields true whereas it fails for the second disjunct. Rule (14b) is treated in the same way where only the test on the second disjunct can be satisfied. Both results are joined together and are associated with the very same disjunction: ($X=1, in(4) \vee X=3, from(4)$).

- (15) a. $meet(1),$
 $subj(1,2), pro(2), num(2,pl)$
 $obj(1,3), colleague(3),$
 $num(3,pl), spec(3,def),$
 $adjn(X,4), obj(4,5), Berlin(5),$
 $(X=1, in(4) \vee X=3, from(4))$

- b. $\left\{ \begin{array}{l} \left[\begin{array}{l} \text{PRED } meet \langle \boxed{2}, \boxed{3} \rangle \\ \text{SUBJ } \boxed{2} \left[\begin{array}{l} \text{PRED } pro \\ \text{NUM } pl \end{array} \right] \\ \text{OBJ } \boxed{3} \left[\begin{array}{l} \text{PRED } colleague \\ \text{NUM } pl \\ \text{SPEC } def \end{array} \right] \end{array} \right] \\ \boxed{4} \text{ADJN } \left\{ \boxed{4} \text{OBJ } \boxed{5} \left[\text{PRED } Berlin \right] \right\} \\ \left(\begin{array}{l} \left\{ \boxed{4} = \boxed{1}, \boxed{4} \left[\text{PRED } in \langle \boxed{5} \rangle \right] \right\} \\ \vee \\ \left\{ \boxed{4} = \boxed{3}, \boxed{4} \left[\text{PRED } from \langle \boxed{5} \rangle \right] \right\} \end{array} \right) \end{array} \right\}$

As a final result we get the packed representation in (15), where the two prepositions are distributed into the local disjunction without converting to disjunctive normal form.

The transferred packed representation corresponds to the two possible utterances in (16). It would be left as a task for the (human) negotiator to find out which of the two sentences would be more appropriate in a given context situation. Due to the local nature of the disjunctions they can be handed over to an additional resolution component in order to disambiguate them or if the discourse and world knowledge is not sufficient for disambiguating to leave them as choices for the human translator.

- (16) a. we will meet the colleagues **in** Berlin
b. we will meet the colleagues **from** Berlin

The main advantage of such an approach is that the transfer rules are independent of the fact whether they are applied to packed representations or not. Unpacking is done only locally and as much as necessary. Only the internal processing needs to be adapted in order to keep track which of the local disjuncts are processed. This is done with a simple book-keeping mechanism which keeps track for any individual term to which local disjunct it belongs. Technically, it is done by using the contexted constraints as described in Maxwell III and Kaplan (1991). Hence the whole mechanism can be kept fully transparent for the transfer rule writer and all of the complexity can be dealt with internally in the transfer rule compiler which compiles the external transfer rule format into an executable Prolog program which propagates the necessary variable sharings.

In order to avoid duplicated work while trying to apply all possible transfer rule combinations the transfer system uses an internal chart to store all successful rule applications. Each predicate in the input set gets assigned a unique bit in a bit vector such that it can be checked easily that no predicate is covered more than once while trying to combine different edges in the chart. With this scheme it is also possible to identify the final edges because they are the ones where all bits are set. The overall processing scheme using an agenda and the data structures are very similar to the chart representation as proposed for doing chart-based generation from ambiguous input (cf. Kay (1996) and Shemtov (1996)). The main difference stems from the lack of explicit context-free grammar rules. Instead, in the proposed setup, the left hand sides

of transfer rules are interpreted as immediate dominance rules as they are used for describing free word order languages supplemented with a single binary context-free rule which recursively tries to combine all possible subsets of terms for which no explicit transfer rule exists.

7 Summary

In this paper we have demonstrated that a Shake-and-Bake inspired MT approach can be applied to flat f-structure representations. It has also been shown how such a transfer system can be combined with the treatment of packed ambiguities for the representation of (syntactic) ambiguities to achieve a truly ambiguity preserving translation architecture. Since the particular treatment of syntactic ambiguities is orthogonal to the possibility of using underspecified semantic representations, the same extension could also be applied for a semantic-based transfer approach on flat representations as advocated for example in Copestake et al. (1995) and Dorna and Emele (1996). The advantage for doing transfer on the level of underspecified semantic representations is the gain of parallelism between source and target language due to the abstraction and underspecification of language specific idiosyncracies which are already dealt with in the linking between syntactic and semantic information. Popular examples are cases of head-switching, category switching and diathesis etc. which disappear on the level of semantic representations (e.g. Dorna et al. (1998)). The discussion of such examples can be found at length in the literature and will therefore not be repeated here.

The proposed transfer architecture is currently being implemented as an extension to an experimental transfer MT system which is fully integrated and interfaced with the XLE system for doing parsing and generation. The application domain comprises the translation of instruction manuals.

References

- Hiyan Alshawi, David M. Carter, Björn Gambäck, and Manny Rayner. 1991. Translation by Quasi Logical Form Transfer. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL'91)*, pages 161–168, Berkeley, CA.

- Hiyan Alshawi, editor. 1992. *The Core Language Engine*. ACL-MIT Press Series in Natural Languages Processing. MIT Press, Cambridge, Mass.
- John Bear and Jerry R. Hobbs. 1988. Localizing expression of ambiguity. In *Proceedings of the 2nd International Conference on Applied Natural Language Processing*, pages 235–241, Texas, Austin.
- J. L. Beaven. 1992. Shake-and-Bake Machine Translation. In *Proceedings of the 14th International Conference on Computational Linguistics (Coling'92)*, Nantes, France.
- J. Bos, B. Gambäck, C. Lieske, Y. Mori, M. Pinkal, and K. Worm. 1996. Compositional Semantics in Verbmobil. In *Proceedings of the 16th International Conference on Computational Linguistics (Coling'96)*, Copenhagen, Denmark.
- A. Copestake, D. Flickinger, R. Malouf, S. Riehemann, and I. Sag. 1995. Translation using Minimal Recursion Semantics. In *Proceedings of the 6th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI'95)*, Leuven, Belgium.
- Michael Dorna and Martin C. Emele. 1996. Semantic-based Transfer. In *Proceedings of the 16th International Conference on Computational Linguistics (Coling'96)*, Copenhagen, Denmark.
- Michael Dorna, Anette Frank, Josef van Genabith, and Martin C. Emele. 1998. Syntactic and semantic transfer with f-structures. In *Proceedings of the 17th International Conference on Computational Linguistics (Coling-ACL'98)*, Montréal, Canada, August.
- Jochen Dörre. 1997. Efficient construction of underspecified semantics under massive ambiguity. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-EACL'97)*, Madrid, Spain.
- M. Egg and K. Lebeth. 1995. Semantic underspecification and modifier attachment ambiguities. In J. Kilbury and R. Wiese, editors, *Integrative Ansätze in der Computerlinguistik. Beiträge zur 5. Fachtagung der Sektion Computerlinguistik der Deutschen Gesellschaft für Sprachwissenschaft (DGfS)*, pages 19–24, Düsseldorf, Germany.
- Mark Johnson. 1991. Features and Formulae. *Computational Linguistics*, 17(2):131–151.
- Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. MIT Press, Cambridge, Mass.
- M. Kay, M. Gawron, and P. Norwig. 1994. *Verbmobil: a Translation System for Face-to-Face Dialogs*. Number 33 in CSLI Lecture Notes. University of Chicago Press.
- Martin Kay. 1996. Chart generation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL'96)*, pages 200–204, Santa Cruz, CA.
- Martin Kay. 1997. The Proper Place of Men and Machines in Language Translation. *Machine Translation*, 12:3–23.
- John T. Maxwell III and Ronald M. Kaplan. 1991. A method for disjunctive constraint satisfaction. In Masaru Tomita, editor, *Current Issues in Parsing Technology*, pages 18–27. Kluwer Academic Publishers, Dordrecht, Holland.
- John T. Maxwell III and Ronald M. Kaplan. 1993. The interface between phrasal and functional constraints. *Computational Linguistics*, 19(4):571–590.
- John T. Maxwell III and Ronald M. Kaplan. 1996. An efficient parser for LFG. In *Proceedings of the 1st LFG Conference*.
- Manfred Pinkal. 1995. Radical Underspecification. In *Proceedings of the 10th Amsterdam Colloquium*, pages 587–606, Amsterdam, Holland, December. ILLC/Department of Philosophy, University of Amsterdam.
- Uwe Reyle. 1993. Dealing with Ambiguities by Underspecification: Construction, Representation and Deduction. *Journal of Semantics*, 10(2):123–179.
- E. Rich, J. Barnett, K. Wittenburg, and D. Wroblewski. 1987. Ambiguity procrastination. In *Proceedings of the 6th National Conference of the American Association for Artificial Intelligence (AAAI'87)*, pages 571–576, Seattle, WA.
- Michael Schiehlen. 1996. Semantic Construction from Parse Forests. In *Proceedings of the 16th International Conference on Computational Linguistics (Coling'96)*.
- Jungyun Seo and Robert F. Simmons. 1989. Syntactic graphs: A representation for the union of all ambiguous parse trees. *Computational Linguistics*, 15(1):19–32, March.
- Hadar Shemtov. 1996. Generation of Paraphrases from Ambiguous Logical Forms. In *Proceedings of the 16th International Conference on Computational Linguistics (Coling'96)*, Copenhagen, Denmark.
- Hadar Shemtov. 1997. *Ambiguity Management in Natural Language Generation*. Ph.D. thesis, Stanford University, June.
- Kees van Deemter and Stanley Peters, editors. 1996. *Semantic ambiguity and underspecification*. Number 55 in CSLI Lecture Notes. CSLI Publications, Stanford University, CA.
- Pete Whitelock. 1992. Shake-and-Bake Translation. In *Proceedings of the 14th International Conference on Computational Linguistics (Coling'92)*, pages 784–791, Nantes, France.