Reversing F-Structure Rewriting for Generation from Meaning Representations

Sina Zarrieß Jonas Kuhn
University of Stuttgart, IMS, Germany

1. Introduction

The standard meaning construction approach for LFG grammars implemented in the XLE framework is the transfer semantics system developed by (Crouch and King, 2006). Their system exploits the XLE transfer module to map LFG f-structures to shallow meaning representations on the basis of an ordered list of term-rewrite rules. A commonly mentioned disadvantage of this transfer approach is the fact that the XLE rewrite rules are not reversible (Crouch and King, 2006).

In this paper, we present a method to automatically induce a transfer grammar which maps meaning representations back to f-structures. The method converts sequences of seen instances of "forward" transfer rules into their approximated "backward" counterparts. The reversal method can be applied to any comparable transfer semantics.The ultimate goal of this work is to have an LFG-based generation system that can generate more grammatical paraphrases than f-structure based generation (e.g. voice alternations). We basically pursue the same strategy as former work in LFG-based generation (e.g. Cahill et al. (2007)), where all possible linguistic realisations of an abstract input representation are first generated and the final output sentence is then selected by a realisation ranking model which is based on linguistic experience and/or trained on a corpus.

2. Main Idea

Intuitively, it is problematic to reverse the f-structure rewriting because the XLE transfer rules can arbitrarily delete information from the input f-structure. As an example, the topmost f-structure in figure 1 contains features like CASE, PERS, or TOPIC which do not appear at all in the meaning representation. Moreover, the node embeddings in meaning representation differ from the f-structure so that we have to be very careful to reintroduce the right f-structure nodes in the reverse mapping.

One of the main ideas of our approach is that we do not need to generate full-fledged f-structures from the meaning representations because the XLE generator can handle underspecified input (Crouch et al., 2004). If the generator does not find a CASE feature in a place where it would expect one, it can freely add all possible values of that feature. The output of the f-structure generation is the complete set of sentences compatible with its (more or less abstract) input and the constraints encoded in the grammar and lexicon.

Thus, the main question is whether we can split up the f-structure features into classes of (a) core meaning features and (b) syntax-internal features; such that only the former need to be present in the semantic input representation. Of course, in principle, it cannot be guaranteed that all the core features are recoverable because the transfer rules may perform arbitrary deletions on the input. However, empirically, our experiments confirm the intuition that the meaning representation generally preserves information about a) the "lexical" features, such that PRED values and argument functions can be reconstructed, b) the semantic features like tense of a verb or definiteness of a noun. An interesting exception to b) is a class of implicit default features in the semantics which are explicit in the f-structure. For instance, in the meaning representation, the clause type is only marked if it is not declarative, whereas the f-structure always marks the clause type. This means that the clause type feature cannot (automatically) be reconstructed for declarative sentences - but if we leave this feature underspecified in the f-structure input, we will always generate questions and declarative sentences for a declarative semantic input. Currently, we deal with this problem by manually configuring the XLE generator in a way such that it only adds certain default values for a class of features which are not specified in the semantic input.

Our implementation reduces the task of inducing a mapping from meaning representations to f-structures to two subproblems: 1) determine the set of features that can actually be recovered from the original f-structure 2) determine the rewrite sequence between a recoverable f-structure term and its corresponding term in the meaning representation. Both of these subproblems cannot be solved by just looking at the "forward" transfer grammar. The reason is that the exact application of the transfer grammar on a given f-structure depends on the "feeding and bleeding" of the single rules. However, given a concrete instance of a <f-structure, meaning representation> pair defined by the transfer mapping, one can exactly tell which f-structure features have been deleted and which features have a correspondence in the meaning representation (in section 4, we will define what "correspondence" formally means). Moreover, it is possible to run the XLE transfer system in a mode that outputs each rule application of the rewrite process. This means that we will be able to recover the sequence of rules which mapped a concrete term in the f-structure to a term in the meaning representation.

The algorithm we implemented approximates the reverse of a given "forward" transfer grammar by operating on instances of transfer rules which have been encountered in a concrete rewrite sequence. If we run it on a large set of <f-structure, rewrite sequence, meaning representation> triples (coming from testsuites or corpus sententences) and take the union[1] of the reversed rules, we can expect all the relevant rules to be recovered. In this way, the method is generally applicable to comparable transfer grammars.

[1] Technically, alternative rules for the same input are allowed by making the rewrite rule applications optional (or at least a subset of them).

3. Example

Once we have induced the reverse transfer grammar, the generation system works as is illustrated in figure 1.

First, an input corpus sentence is parsed and mapped to a flat semantic representation. Note that the subject of the passive f-structure is mapped to a "semantic object" in the meaning representation. If our reversal method has encountered rewrite sequences for active and passive sentences, it will know that both subjects of active f-structures and objects of passive f-structures are mapped to "semantic objects". In the reverse mapping from meaning representation to f-structures, it will thus produce an f-structure chart that, besides the original f-structure, contains its paraphrase f-structure in active voice.

'Tom was discovered by Chomsky.'

↓ *XLE parsing*

Original F-Structure:

PRED	'discover < (↑ SUBJ)(↑ OBL-AG) >'
SUBJ	PRED 'Tom'; NTYPE proper-name; PERS 3
OBL-AG	PRED 'Chomsky'; ...
TOPIC	['Tom']
VTYPE	main
PASSIVE	+

↓ *transfer meaning construction*

HEAD (discover)
PAST (discover)
ROLE (sem-subj,discover,Chomsky)
ROLE (sem-obj,discover,Tom)

↓ *reverse transfer mapping*

Output F-Structure Chart:

PRED	'discover < (↑ SUBJ)(↑ OBJ) >'
SUBJ	PRED 'Chomsky'
OBJ	PRED 'Tom'
PRED	'discover < (↑ SUBJ)(↑ OBL-AG) >'
SUBJ	PRED 'Tom'
OBL-AG	PRED 'Chomsky'

↓ *XLE generation*

"Tom was discovered by Chomsky."
"Chomsky discovered Tom."

↓ *surface realisation ranking*

"Tom was discovered by Chomsky."

Figure 1: Generation via meaning representations

Also note that the generated f-structures are much more underspecified than the original one because e.g. TOPIC or CASE cannot be recovered from the semantic representation. The XLE generator adds these missing features and outputs all strings which are proper linguistic realisations of the generated f-structure chart.

The resulting set of strings is passed to a generation ranking module which selects the most appropriate paraphrase realisation in the given context (Cahill et al., 2007).

4. Instance-based Transfer Reversal

In section 2, we pointed out that we can partially reverse instances of f-structure rewrite *sequences*. We will briefly sketch this intuition here, presupposing some basic familiarity with the transfer mechanism.

The XLE transfer system represents f-structures as sets of two-place predicates. The predicate's name represents the f-structure attribute, the first argument its f-structure node and the second argument the embedded f-structure node or feature value (e.g. CASE(var(1),acc) or OBJ(var(0),var(1))). We define a rewrite mapping to be reversable if the predicate name and the atomic values of the left-hand term have a correspondence in the right-hand term. For instance, "CASE(%Var,%Value) ==> noun(%Var)" is not reversable because the second argument of the CASE predicate is deleted. On the other hand, "PRED(%Var,%Word) ==> word(%Word)" is reversable, because it only deletes the f-structure node. In this way, we allow the reverse transfer rules to reintroduce f-structure nodes. Now, an input f-structure predicate x can be defined as reversable if there is a sequence of reversable mappings between x and an output semantic clause y.

To prevent the reverse transfer rules from introducing arbitrary f-structure nodes, we have to add conditions to the left-hand side of the rules. These conditions can also be inferred automatically from seen instances of rewrite sequences. For instance, given the reverse instances "word('do') ==> PRED(var(0),'do'), past('do') ==> TENSE(var(0),past)", we know that the PRED clause has to be a condition for the second rule. Otherwise, the rule sequence "word(%Word) ==> PRED(%Word,%Var), past(%Word) ==> TENSE(%Var,past)" would introduce two independent f-structure nodes.

References

Cahill, Aoife, Martin Forst, and Christian Rohrer. 2007. Stochastic realisation ranking for a free word order language. In *Proceedings of the 11th European Workshop on NLG*, pp. 17–24, Saarbrücken, Germany.

Crouch, Dick, and Tracy Holloway King. 2006. Semantics via F-Structure Rewriting. In Miriam Butt and Tracy Holloway King (eds.), *Proceedings of the LFG06 Conference*.

Crouch, Richard, Tracy Holloway King, John T. Maxwell III, Stefan Riezler, and Annie Zaenen. 2004. Exploiting F-structure Input for Sentence Condensation . In Miriam Butt and Tracy Holloway King (eds.), *Proceedings of the LFG04 Conference*, University of Canterbury.