Generation and synchronous tree-adjoining grammars

Generation and Synchronous Tree-Adjoining Grammars

Stuart M. Shieber
Aiken Computation Laboratory
Division of Applied Sciences
Harvard University
Cambridge, MA 02138

Yves Schabes
Department of Computer and
Information Science
University of Pennsylvania
Philadelphia, PA 19104

Abstract

Tree-adjoining grammars (TAG) have been proposed as a formalism for generation based on the intuition that the extended domain of syntactic locality that TAGs provide should aid in localizing semantic dependencies as well, in turn serving as an aid to generation from semantic representations. We demonstrate that this intuition can be made concrete by using the formalism of synchronous tree-adjoining grammars. The use of synchronous TAGs for generation provides solutions to several problems with previous approaches to TAG generation. Furthermore, the semantic monotonicity requirement previously advocated for generation grammars as a computational aid is seen to be an inherent property of synchronous TAGs.

Introduction

The recent history of grammar reversing can be viewed as an effort to recover some notion of semantic locality on which to base a generation process. For instance, Wedekind (1988) requires a property of a grammar that he refers to as *connectedness*, which specifies that complements be semantically connected to their head. Shieber (1988) defines a notion of *semantic monotonicity*, a kind of compositionality property that guarantees that it can be locally determined whether phrases can contribute to forming an expression with a given meaning. Generation schemes that reorder top-down generation (Dymetman and Isabelle, 1988; Strzalkowski, 1989) so as to make available information that well-founds the top-down recursion also fall into the mold of localizing semantic information. Semantic-head-driven generation (Shieber *et al.*, forthcoming; Calder *et al.*, 1989) uses semantic heads and their complements as a locus of semantic locality.

Joshi (1987) points out that tree-adjoining grammars may be an especially appropriate formalism for generation because of their syntactic locality properties, which, intuitively at least, ought to correlate with some notion of semantic locality. The same observation runs as an undercurrent in the work of McDonald and Pustejovsky (1985), who apply TAGs to the task of generation. As these researchers note, the properties of TAGs for describing the syntactic structuring of a natural language mesh quite naturally with the requirements of natural-language generation. Nonetheless, generation is not, as typically viewed, a problem in natural-language syntax. Any system that attempts to use the TAG formalism as a substrate upon which to build a generation component must devise some mechanism by which a TAG can articulate appropriately with semantic information. In this paper, we discuss one such mechanism, synchronous TAGs, which we have previously proposed in the arena of semantic interpretation and automatic translation, and examine how it might underly a generation system of the sort proposed by Joshi and McDonald and Pustejovsky. In particular, synchronous TAGs allow for a precise notion of semantic locality corresponding to the syntactic locality of pure TAGs.

Scope of the Paper

The portion of the full-blown generation problem that we address here is what might be referred to as the tactical as opposed to the strategic generation problem. That is, we are concerned only with how to compute instances of a well-defined relation between strings and canonical logical forms[1] in the direction from logical forms to strings, a problem that is sometimes referred to as "reversing" a grammar. This aspect of the generation problem, which ignores the crucial issues in determining what content to communicate, what predicates to use in the communication, and so forth, can be seen as the reverse of the problem of parsing natural language to derive a semantic representation. The citations in the first paragraph can serve to place the issue in its historical research context. The other truly difficult issues of general natural-language production are well beyond the scope of this paper.

[1]This issue of canonicality of logical forms is discussed by Shieber (1988).

Semantics in Generation

Although Joshi discusses at length the properties of TAGs advantageous to the generation task (1987), he does not address the issue of characterizing a semantic representation off of which generation can proceed. McDonald and Pustejovsky do mention this issue. Because TAGs break up complex syntactic structures into elementary structures in a particular way, their semantic representation follows this structuring by breaking up the logical form into corresponding parts. McDonald and Pustejovsky consider the sentence

(1) How many ships did Reuters report that Iraq had said it attacked?

Its semantic representation follows the decomposition of the sentence into its elementary TAG trees—corresponding (roughly) to "How many ships ...it attacked", "did Reuters report that ...", "Iraq had said ...". McDonald and Pustejovsky describe their semantic representation: "The representation we use ... amounts to breaking up the logical expression into individual units and allowing them to include references to each other." The units for the example at hand would be:

$$
\begin{aligned}
U_1 &= \lambda(\textit{quantity-of-ships}).\\
&\qquad \textit{attack}(\textit{Iraq}, \textit{quantity-of-ships})\\
U_2 &= \textit{say}(\textit{Iraq}, U_1)\\
U_3 &= \textit{report}(\textit{Reuters}, U_2)
\end{aligned}
$$

By composing the units using substitution of equals for equals, a more conventional logical form representation is revealed:

$$
\begin{aligned}
&\textit{report}(\textit{Reuters},\\
&\quad \textit{say}(\textit{Iraq},\\
&\qquad \lambda(\textit{quantity-of-ships}).\\
&\qquad\quad \textit{attack}(\textit{Iraq}, \textit{quantity-of-ships})))
\end{aligned}
$$

Three problems present themselves.

First, the particular decomposition of the full semantic form must be explicitly specified as part of the input to the generation system.

Second, the basic operation that is used (implicitly) to compose the individual parts, namely *substitution* does not parallel the primitive operation that TAGs make available, namely *adjunction*. In the particular example, this latter problem is revealed in the scope of the quantity quantifier being inside the *say* predicate. The more standard representation of scoping would be akin to

$$
(2)\quad
\begin{aligned}
&\lambda(\textit{quantity-of-ships}).\\
&\quad \textit{report}(\textit{Reuters},\\
&\qquad \textit{say}(\textit{Iraq}, \textit{attack}(\textit{Iraq},\\
&\qquad\qquad \textit{quantity-of-ships})))
\end{aligned}
$$

but this requires one of the elementary semantic units to be "broken up". Consequently, McDonald and Pustejovsky note that they cannot have the logical form (2) as the source of the example sentence (1).

Third, the grammatical information alone does not determine where adjunctions should occur. McDonald and Pustejovsky allude to this problem when they note that "the [generator] must have some principle by which to judge where to start." In their own example, they say that "the two pending units, U_2 and U_3, are then attached to this matrix ... into complement positions," but do not specify how the particular attachment positions within the elementary trees are chosen (which of course has an impact on the semantics). The relationship between syntax and semantics that they propose links elementary trees with units of the realization specification. Apparently, a more finely structured representation is needed.

Synchronous TAGs

In order to provide an explicit representation for the semantics of strings generated by a TAG, and in so doing provide a foundation for the generation efforts of Joshi and McDonald and Pustejovsky, we present an extension to TAGs, synchronous TAGs, which was originally developed just to characterize the declarative relationship between strings and representations of their semantics. The formalism allows us to circumvent some of the problems discussed above.

The idea underlying synchronous TAGs is simple. One can characterize both a natural language and a logical form language with TAGs. The relation between strings in the two languages (sentences and logical forms, respectively) can then be rigorously stated by pairing the elementary trees of the two grammars and linking the corresponding nodes, forming a new grammar whose elements are linked pairs of elementary trees.

The synchronous TAG formalism addresses all three of the problems mentioned above. First, a synchronous TAG characterizes a relation between languages. Thus, we need not assume that the sentences of the logical form language come pre-packaged into their constituent units (just as in the case of sentence parsing, where we need not assume that sentences come pre-bracketed). Second, the operations that are used to build the two structures—natural language sentences and semantic representations—are stated using the same kinds of operations, as they are both characterized by TAGs. Third, the linking of individual nodes in the elementary trees of a synchronous TAG provides just the fine-grained relationship between syntax and semantics that allows decisions about where to perform semantic operations to be well-defined.

An Example Synchronous TAG

We introduce synchronous TAGs by example, continuing with an exegesis of the sentence that McDonald and Pustejovsky focus on, and following roughly the

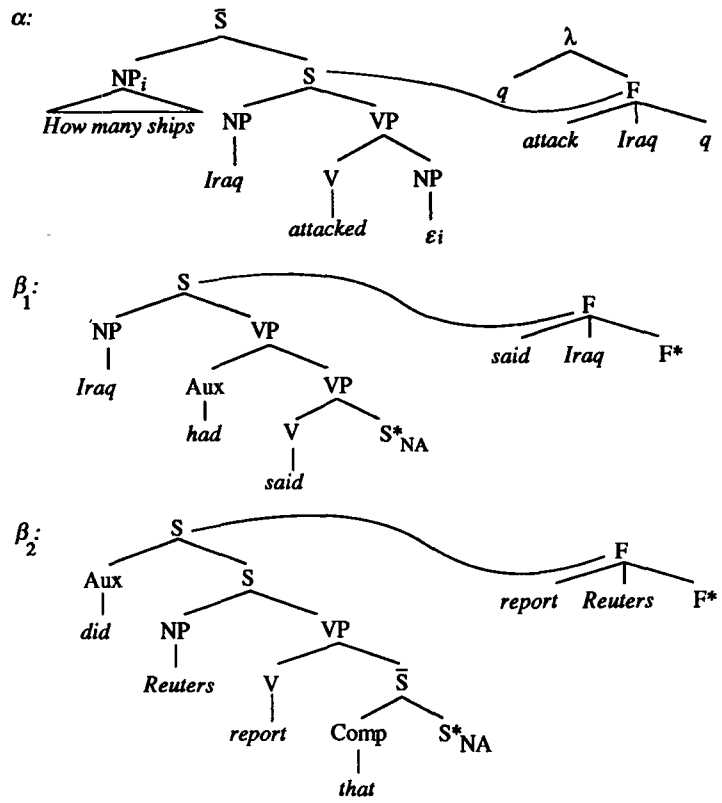

Figure 1: Example Synchronous TAG

structure of their TAG analysis.[2]

A synchronous TAG sufficient for this example includes the three pairings of trees (labeled α, β_1, and β_2) found in Figure 1. Note that the first components of the three pairs constitute a TAG grammar sufficient to generate the sentence "How many ships did Reuters report that Iraq attacked" or "How many ships did Reuters report that Iraq said that Iraq attacked". The second components generate strings in a logical form language. The syntax of that language includes such phrase types as formula (F) or abstracted property (λ). The obvious linearization of such trees will be assumed, so that the logical form in given for the sample sentence is in the language. Some of the nodes in the pairs are linked; formally, the interpretation of these links is that operations on the tree pairs must occur at both ends of a link. For simplicity, we have marked only those links that will be needed for the derivation of the sample sentence.

Derivation in the synchronous grammar proceeds by choosing a pairing of initial trees from the grammar and repeatedly updating it by the following three-step process:[3]

1. Choose a link to act upon.
2. Choose a pairing such that the two trees can respectively act on (substitute at or adjoin at) the respective ends of the link chosen in Step 1.
3. Remove the chosen link from the trees being updated and perform the two operations, one in each of the trees. If the trees in the chosen pairing themselves have links, these are preserved in the result.

For instance, we might start with the initial tree pair α from Figure 1. We choose the sole link in α, and choose β_1 as the tree pair to operate with, as the first component of β_1 can operate (by adjunction) on an S

[2] The linguistic analysis implicit in the TAG English fragment that we present is not proposed as an appropriate one in general. It merely provides sufficient structure to make the points vis-à-vis generation. Furthermore, the trees that we present here for expository purposes as elementary should actually themselves be built from more primitive trees. Finally, we gloss over details such as features necessary to control for agreement or verb-form checking, and we replace the pronoun with its proper noun antecedent to finesse issues in pronominal interpretation.

[3] A fuller description of the formal aspects of synchronous TAGs can be found in a previous paper (Shieber and Schabes, forthcoming).

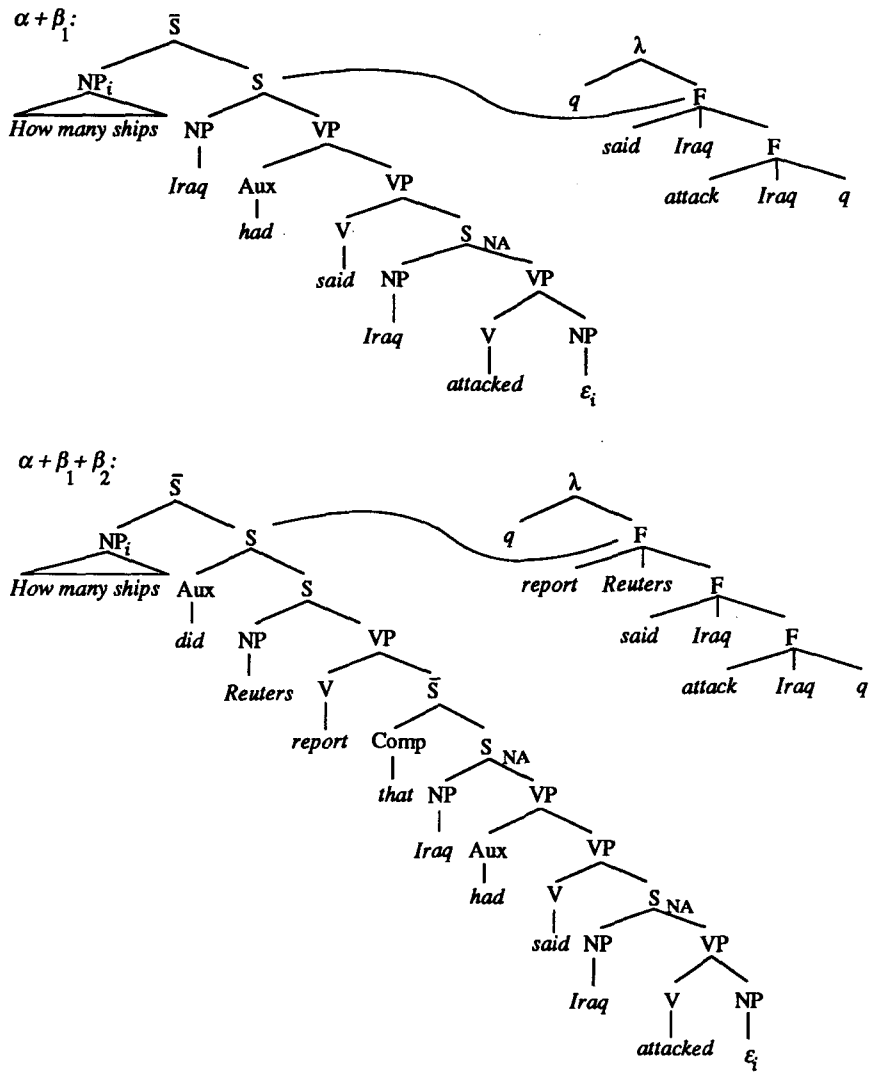

Figure 2: Results of Synchronous Derivation Steps

node, and the second on an F node as required by the chosen link. The result of performing the adjunctions is the pairing given as $\alpha + \beta_1$ in Figure 2. The link in the β_1 pair is preserved in the resultant, and can serve as the chosen link in the next round of the derivation. This time, we use β_2 to operate at each end of the link resulting in the pairing labeled $\alpha + \beta_1 + \beta_2$. This pairing manifests the association between the English string "How many ships did Reuters report that Iraq said that Iraq attacked" and the logical form representation in (2).

Returning to the three issues cited previously, the synchronous TAG presented here:

1. Makes the decomposition of the logical forms implicit in the grammar just as the decomposition of the natural-language expressions are, by stating the structure of logical forms grammatically.
2. Allows the same operations to be used for composing both natural-language expressions and semantic representations as both are stated with the same grammatical tools.
3. Makes the fine-grained correspondance between expressions of natural language and their meanings explicit by the technique of node linking.

The strong notion of semantic locality that synchronous TAGs embody makes these results possible. This semantic locality, in turn, is only possible because the extended domain of locality found in pure TAGs makes it possible to localize dependencies that would otherwise be spread across several primitive structures.

Translation with Synchronous TAGs

Synchronous TAGs as informally described here declaratively characterize a relation over strings in two languages without priority of one of the languages over the other. Any method for computing this relation in one direction will perforce be applicable to the other direction as well. The distinction between parsing and generation is a purely informal one depending merely on which side of the relation one chooses to compute from; both are instances of a process of translating between two TAG languages appropriately synchronized.

The question of generation with synchronous TAGs reverts then to one of whether this relation can be computed in general. There are many issues involved in answering this question, most importantly, what the underlying TAG formalism (the *base* formalism) is that the two linked TAGs are stated in. The simple example above required a particularly simple base formalism, namely pure TAGs with adjunction as the only operation. The experience of grammar writers has demonstrated that substitution is a necessary operation to be added to the formalism, and that a limited form of feature structures with equations are helpful as well. Work on the use of synchronous TAGs to capture quantifier scoping possibilities makes use of so-called multi-component TAGs. Finally, the base TAGs may be lexicalized (Schabes *et al.*, 1988) or not.

Once the base formalism has been decided upon (we currently are using lexicalized multi-component TAGs with substitution and adjunction), a simple translation strategy from a source string to a target is to parse the string using an appropriate TAG parser for the base formalism. Each derivation of the source string can be mapped according to the synchronizing links in the grammar to a target derivation. Such a target derivation defines a string in the target language which is a translate of the source string.

In the case of generation, the source string is a semantic representation, the target is a natural-language realization. For example, the logical form (2) has a single derivation in the pure TAG formed by projecting the synchronous TAG onto its semantic component. (We might notate the semantic components with α(sem), β_1(sem), and β_2(sem), and analogously for the syntactic components.) That derivation can be recovered by "parsing" the logical form with the projected logical form grammar, as depicted in Figure 3. The pairings whose semantic components were used in this derivation and the links operated on implicitly define a corresponding derivation on the syntactic side. The yield of this derivation is a string whose meaning is represented by the logical form that we started with.

The target derivation might not, unlike in the example above, be in canonical form (as defined by Vijay-Shanker (1988)), and consequently must be normalized to put it into canonical form. Under certain configurations of links, the normalization process is nondeterministic; thus one source derivation (necessarily in canonical form by virtue of properties of the parsing algorithm) may be associated with several canonical target derivations. In translation from natural language to logical forms, the multiple translates typically correspond to scope ambiguities in the source sentence (as quantifier scope or scope of negation or adverbs). On the other hand, we have not observed the linking configurations that give rise to such ambiguities in translating in the other direction, that is, in performing generation.

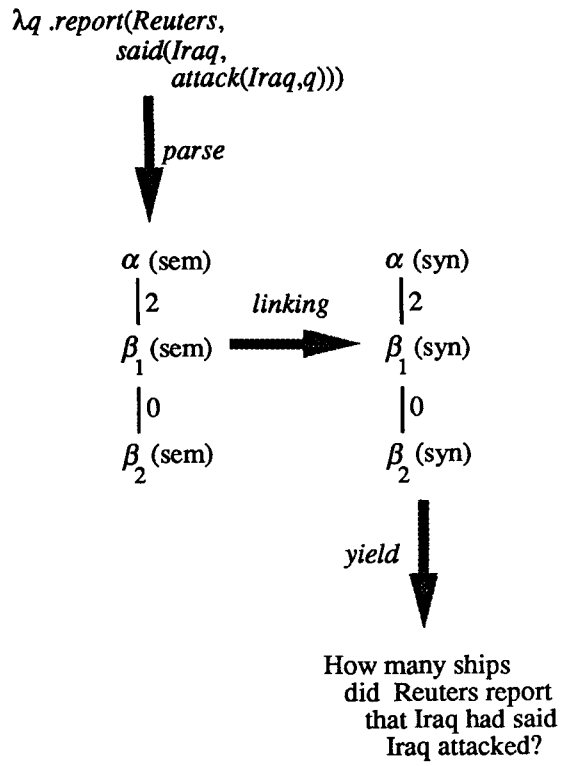

Figure 3: Generation by Derivation Translation

In previous work, one of us noted that generation according to an augmented context-free grammar can be made more efficient by requiring the grammar to be *semantically monotonic* (Shieber, 1988); the derived semantics for an expression must include, in an appropriate sense, the semantic material of all its subconstituents. It is interesting to note that synchronous TAGs are inherently semantically monotonic, and the computational advantages that accrue to such grammars apply to synchronous TAG generation as well. Furthermore, it is reasonable to require that the semantic component of a synchronous TAG be *lexicalized* (in the sense of Schabes et al. (1988)), allowing for more efficient parsing according to the semantic grammar and, consequently, more efficient generation. In the case of augmented context-free grammars, the semantic monotonicity requirement precludes "lexicaliza-

tion" of the semantics. It is not possible to require nontrivial semantics to be associated with each lexical item. This fact, and the inefficiencies of generation thatfollow from it, was the initial motivation for the move to semantic-head-driven generation (Shieber *et al.*, forthcoming). The efficiencies that that algorithmgains for augmented-context-free generation inhere in the synchronous TAG generation process if the semantic gramamr is lexicalized. In summary, just as lexicalization of the syntactic grammar aids parsing (Schabes and Joshi, 1989), so lexicalization of the semantic grammar aids generation.

The simple generation algorithm that we have just presented seems to require that we completely analyze the logical form before generating the target string, as the process is a cascade of three subprocesses: parsing the logical form to a source derivation, mapping from source to target derivation, and computing the target derivation yield. As is common in such cases, portions of these computations can be interleaved, so that generation of the target string can proceed incrementally while traversing the source logical form. To what extent this incrementality can be achieved in practice depends on subtleties in the exact formal definition of synchronous TAG derivation and properties of particular grammars; a full explication is beyond the scope of this paper.

Conclusion

The extended domain of locality that tree-adjoining grammars enjoy would seem to make them ideal candidates for the task of tactical generation, where semantic locality is of great importance. Synchronous TAGs, which extend pure TAGs to allow for mappings between languages, provide a formal foundation for this intuition by making explicit the semantic locality that generation requires.

Acknowledgements

This research was partially funded by ARO grant DAAG29-84-K-0061, DARPA grant N00014-85-K0018, and NSF grant MCS-82-19196 at the University of Pennsylvania. We are indebted to Aravind Joshi for his support of this research.

Bibliography

Jonathan Calder, Mike Reape, and Hank Zeevat. An algorithm for generation in unification categorial grammar. In *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics*, pages 233–240, Manchester, England, 10-12 April 1989. University of Manchester Institute of Science and Technology.

Marc Dymetman and Pierre Isabelle. Reversible logic grammars for machine translation. In *Proceedings of the Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Pittsburgh, Pennsylvania, 1988. Carnegie-Mellon University.

Aravind K. Joshi. The relevance of tree adjoining grammar to generation. In Gerard Kempen, editor, *Natural Language Generation*, chapter 16, pages 233–252. Martinus Nijhoff Publishers, Dordrecht, Holland, 1987.

David D. McDonald and James D. Pustejovsky. TAGs as a grammatical formalism for generation. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, pages 94–103, University of Chicago, Chicago, Illinois, 8-12 July 1985.

Yves Schabes and Aravind K. Joshi. The relevance of lexicalization to parsing. In *Proceedings of the International Workshop on Parsing Technologies*, pages 339–349, Pittsburgh, Pennsylvania, 28–31 August 1989. Carnegie-Mellon University.

Yves Schabes, Anne Abeillé, and Aravind K. Joshi. Parsing strategies with 'lexicalized' grammars: Application to tree adjoining grammars. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING'88)*, Budapest, August 1988.

Stuart M. Shieber and Yves Schabes. Synchronous tree-adjoining grammars. In *Proceedings of the 13th International Conference on Computational Linguistics*, University of Helsinki, Helsinki, Finland, forthcoming.

Stuart M. Shieber, Gertjan van Noord, Fernando C. N. Pereira, and Robert C. Moore. Semantic-head-driven generation. *Computational Linguistics*, forthcoming.

Stuart M. Shieber. A uniform architecture for parsing and generation. In *Proceedings of the 12th International Conference on Computational Linguistics*, pages 614–619, Karl Marx University of Economics, Budapest, Hungary, 22–27 August 1988.

Tomek Strzalkowski. Automated inversion of a unification parser into a unification generator. Technical Report 465, Department of Computer Science, New York University, New York, New York, 1989.

K Vijay-Shanker. *A Study of Tree Adjoining Grammars.* PhD thesis, University of Pennsylvania, Philadelphia, Pennsylvania, 1988.

Jürgen Wedekind. Generation as structure driven derivation. In *Proceedings of the 12th International Conference on Computational Linguistics*, pages 732–737, Budapest, Hungary, 1988.