STINA:

A PROBABILISTIC PARSER FOR SPEECH RECOGNITION

Rolf Carlson and Sheri Hunnicutt[*]

Department of Speech Communication and Music Acoustics, KTH

A natural language component for our speech recognition system is under development. Work to date has produced a question parser called STINA which is implemented as a context free grammar compiled into an augmented transition network. Features of STINA are a stack decoding search strategy, a feature-passing mechanism to implement unification and a sentence generation capability; the program is fast (40 ms/sentence) and efficient.

INTRODUCTION

During the past year, we have begun to develop a natural language component for our speech synthesis and speech recognition systems (Blomberg 1992, and Elenius & Takacs 1990). Our initial work has been with a sublanguage grammar, a grammar limited to a particular subject domain, that of requesting information from tables about transportation. The need for this grammar became evident in our plans to develop an experimental system to demonstrate our speech recognition work. We wanted to be able to provide some syntactic and semantic knowledge to the recognizer in order to aid in choosing the correct input question from a list of possible questions given as output by the recognizer or to make possible top-down hypotheses. We also wanted a component to aid in accessing information from a data base to answer the question.

Our goals are not to develop a new linguistic theory, and are not to do advanced linguistic research. But we do have the goals of developing a parser that is technically robust -- a parser that is efficient and fast, that is statistically sound, and that fails gracefully. We have also stressed interactive development in order to have control over the system´s progress as more components are added.

METHOD

It was decided to develop our parser along the same lines as a parser used at M.I.T. for a similar purpose. Thus, our fundamental concepts are taken from TINA, a parser developed by Stephanie Seneff for the airline traffic information system, ATIS, which is run as a speech recognition demonstration system. Our parser is called STINA, Swedish TINA. STINA is knowledge-based and is designed as a probabilistic language model. It contains a context free grammar which is compiled into an augmented transition network (ATN). Training the parser by submitting sentences to it for parsing results in arcs of the ATN being labelled with the probability that the parse will transverse them. During parsing, the method of unification is employed for such needs as agreement of features and handling of semantic constraints. Both syntactic and semantic features are defined.

[*] Names in alphabetic order.

Development of the Grammar and Algorithm

The context free grammar was constructed based on a set of 55 example sentences. It gives a single correct parse for all but a few of these sentences. The next step in the development of the parser will be to collect a large amount of data in a "Wizard of Oz" paradigm with the actual speech recognition algorithm uncoupled. This will allow us to test the grammar and parsing algorithm without initially being encumbered by multiple possible text strings. Subjects will be asked to direct questions to the system about a transportation task. Using this new data, successive improvements to the grammar and algorithm will be made. At the same time, probabilities on the arcs in the ATN will be updated by the additional input.

Implementation

The parser is implemented with a stack decoding search strategy based on the TINA principles. After the dynamic network structure of the ATN is created, parse nodes are linked together. They are then placed in a stack according to their probabilities, and taken from the stack one at a time to investigate a possible match with the input text. Different hypotheses can be processed in parallel or the most probable hypothesis can be explored until it successfully reaches the end of the sentence or fails.

There is also a feature-passing mechanism to implement unification in the grammar. This mechanism permits the handling of long-distance grammatical movements, agreement of features and semantic constraints. The constraints should direct the model to not only parse correct sentences, but also to reject incorrect parses or to regard them as unlikely. The rejection mechanism of a parser is of major importance for the speech recognition system.

An important feature which aids in restricting parses to those which are probable is the sentence generation capability. Given the lexicon and grammar, allowable sentences can be generated for inspection. Unlikely or erroneous parses can thus be identified and the offending grammatical or lexical structures modified.

Lexicon

Our lexicon was compiled from all words in the example sentences, and their entries generated by processing them in the Two-Level Morphology (TWOL) lexical analyzer of Koskenniemi (1983). Each entry was then corrected by removing all unknown homographs. New grammatical and semantic features which are used by our algorithm and special application were then added. Since this process can result in two homographs with identical entries, homographs with identical features are deleted at compile time.

Node Matrix

Possible transitions within the grammar are represented by a two-dimensional matrix for each parent node. All transitions at each node are initially assigned equal probability, and are later adjusted by submission of data to the parser (training).

Information Access

Each parse hypothesis is related to a node which is copied each time the hypothesis is advanced. Since all needed information is stored in each parse node on the stack, backtracking is not necessary and the needed control structure is already in place for parallel processing in the future. This makes the system fast and simple.

Setting and Transport of Features; Unification

Features are by default transported from parents to children and between siblings. However, the grammar can force features to be transported up in the tree for unification. Features can also be blocked and changed by certain nodes.

When a possible parse should be analyzed at a terminal (word) node, several mechanisms are activated. The first basic test is to check whether the features of the word agree with the search features. A feature specification is created based on both the search pattern and the word pattern. In order to assure agreement of features at appropriate parse nodes, features which are cumulatively set on the current path are checked for agreement with this pattern. Only those features specified in the unification set (also specified by the grammar) have to agree with the cumulative specification stored in the parse node.

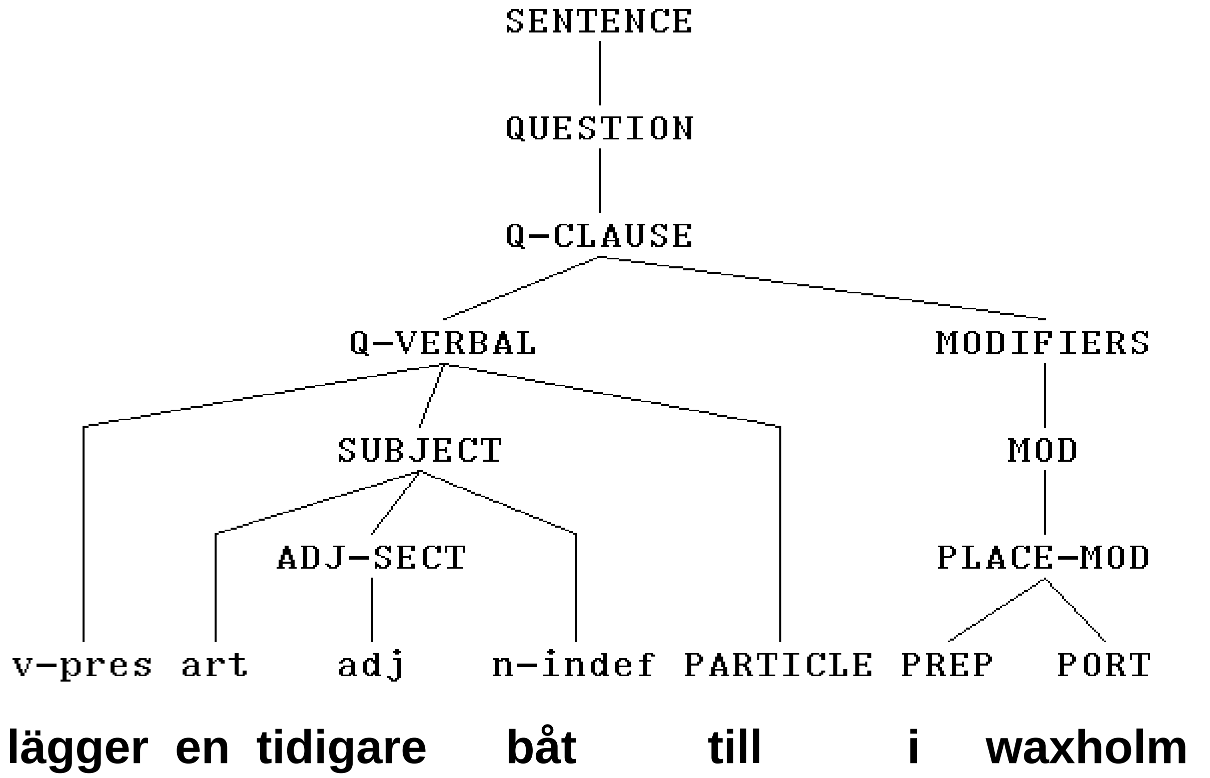

Figure 1. Example of parse tree with long-distance grammatical movement. See text for details.

The example in Figure 1 shows how some of these feature agreements are handled. When the grammatical structure in which "v-pres" is a subnode of "Q-verbal" is hypothesized, the word "lägger" is accepted. It has a feature "+PARTICLE" which is transported up to "Q-Verbal" indicating that a particle can follow it, not necessarily immediately. When the word "till" is later encountered, it is accepted as a particle which, together with "lägger" and a "Subject" node makes up the "Q-verbal." Agreement within the subject is handled similarly, although the elements are adjacent. The indefinite feature on "en" agrees with at least one definition of "Subject" in the grammar, and this feature is transported up to later be matched with the features of "tidigare" (which is both definite and indefinite) and "båt." A semantic feature is also included in this example, the feature "+PORT" on the word "Waxholm." "Port" is a possible transition from "preposition" with parent node "Place-Mod." The unification is here done with the node name itself and no special features need to be tested. The separation between node names and features is done automatically at the time the grammar is compiled.

Generators, Absorbers and Activators

Questions such as "Which train shall I take?" can be considered to be the result of a question formation procedure on the statement "I shall take (a certain) train." This is sometimes annotated with the "trace" convention as "Which train shall I take (t_i)?" where (t_i) denotes the trace of the previous position of the object, "(a certain) train." This type of question is handled in STINA, as it is in TINA, with the concepts "current focus" and "float object" together with the special parse node designations "generate," "absorb" and "activate." Briefly described, the current focus is the most recently mentioned phrase marked by a generator. The parser must then find a grammatical rule containing a node that could have been its trace (the absorber). An activate node which is parent to the absorbing node moves the current focus to the float object which is absorbed.

Some examples from our current task

The task we have set for ourselves deals with public boat traffic in the archipelago. Below are some of the sentences successfully given a single parse by STINA:

Vilka turer går till möja från waxholm?
Hur åker man för att komma till sandhamn från stavsnäs?
Hur många turer går det om dan från stockholm till nämndö?
Vilka båtar har servering ombord?
Finns det någon livsmedelsbutik på möja?
Är båten till sandhamn en ångbåt?
Hur många bryggor stannar båten vid på vägen till grinda?
Lägger båten till i grinda?
Kan man få mat på båten?
Är matserveringen öppen före waxholm?

CONCLUSIONS

In the current functioning system we have an interactive development facility including computer-generated parse trees, and efficient data structures which ensure speedy parsing. Currently, the parser requires only about 40 ms per sentence. It needs only 10 kBytes of data memory and reuses links so that no "garbage collection" is necessary. Statistics are collected automatically.

ACKNOWLEDGEMENTS

We thank Stephanie Seneff for guidance in the implementation and adaptation of this new parser. This work was partially supported by grants from the Swedish Language Technology Programme.

REFERENCES

Blomberg, M. and Elenius, K. (1992): "Continuous speech recognition using synthetic word and triphone prototypes", this conference.

Elenius K. and Takacs, G. (1990): "Acoustic-phonetic recognition of continuous speech by artificial neural networks", in STL-QPSR 2-3 1990.

Koskenniemi, K. (1983): "Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production", University of Helsinki, Department of General Linguistics, Publications No. 11.

Seneff, S. "TINA (*submitted for publication*): A Natural Language System for Spoken Language Applications".

Seneff, S. "TINA (1989): A Probabilistic Syntactic Parser for Speech Understanding Systems," Proceedings ICASSP-89, pp. 711-714.