Lexical Functional Grammar in Speech Recognition

Klaus - Jürgen Engelberg
Fraunhofer Institut IAO
Holzgartenstr. 17
D - 7000 Stuttgart
West Germany

Abstract

The syntax component of the speech recognition system IKAROS¹ is described. The usefulness of a probabilistic Lexical Functional Grammar both for constraining bottom-up hypotheses and top-down predicting is shown.

1. Introduction

The most important problem in all speech recognition systems is the inherent uncertainty associated with the acoustic-phonetic decoding process at the basis of such a system. One approach taken in many existing system to overcome these difficulties is to integrate higher level knowledge sources that have a certain a-priori knowledge about specific problem areas. Following this line of thought, the system architecture adopted in the IKAROS-project assumes different levels of knowledge (representations) e.g. acoustic parameters, phonemes, words, constituent structures etc. The interaction between these knowledge sources is controlled by a central blackboard control module (like in HEARSAY II). This whole system is embedded in an object-oriented environment and communication between the modules is realized by message passing.

Within IKAROS particular attention is given to the problem of using the same knowledge representations both for data-driven bottom-up hypothesizing and expectation-driven top-down prediction and to the problem of providing a general framework of uncertainty management. According to this rationale, the main purpose of the syntax component is to constrain the number of word sequences to be dealt with in the recognition process and to predict or insert poorly recognized words. Grammaticalness in itself is of no importance to us. Quite to the contrary, in a real-live application a certain degree of error tolerance is a desired effect.

In the syntax component of IKAROS we work within the formal framework of a probabilistic Lexical Functional Grammar. Certain modifications to the formalism as expounded in /Bresnan1982/ have been made to suit our purposes. We use as an implementation an event-driven chart-parser that is capable of all the necessary parsing strategies i.e. top-down, bottom-up and left-to-right and right-to-left parsing.

2. Probabilistic context-free Grammars

2.1. The event-driven parser

The interaction between the blackboard manager and the syntax component is roughly as follows: the blackboard manager sends a message to the syntax component indicating that a particular word has been recognized (or rather "hypothesized") at a certain position in the input stream (or in chart-parser terminology with starting and ending vertex number) together with a certain numerical confidence score. The syntax component accumulates information about these (in arbitrary order) incoming word hypotheses and in turn posts hypotheses about predicted and recognized words or constituents on the blackboard. The job of the syntax component now is to decide between several conflicting (or competing) constituent structures stored in the chart i.e. to choose the best grammatical structure.

2.2. The formalism

We assume a probabilistic context-free grammar $G = \langle V_N, V_T, R, S \rangle$:

V_N denotes the nonterminal vocabulary
Nonterminals are denoted by A, B, C ...,
strings of these by X, Y, Z...
lexical categories by P, Q, ...

V_T denotes the terminal vocabulary
terminals (words) denoted by a, b, c, ... ,
strings of both types of symbols are
denoted by w, x, y, z .

R denotes the set of rules $\{R_1, R_2, \dots, R_i\}$
with each rule having the format
 $R_i = \langle A_i \rightarrow X_i, q_i \rangle$
where q_i indicates the a-priori

¹ Research in IKAROS is partially funded by the ESPRIT programme of the European Community under contract P954

probability for the application of this rule

S denotes the initial symbol

Lexical rules have the format

$$L_j = \langle A_j \rightarrow t_j, q_j \rangle$$

In a probabilistic grammar, there is no clearcut dichotomy between grammatical and ungrammatical sentences. Rather, we can devise our language model in such a way that more frequent phrases receive a higher probability than less frequent ones. Even different word orders will have different probabilities.

Now we are able to compute the a-priori probability of a (partial) derivation T starting with the symbol S in the following recursive manner :

$$p(S \leftarrow S) = 1$$

$$p(xYz \leftarrow T \leftarrow S) = p(xAy \leftarrow S) * q, \text{ if there is a rule } \langle A \rightarrow Y, q \rangle \text{ in } R$$

In our implementation, these a-priori probabilities are weighted with the scores delivered for individual words by the acoustic-phonetic component to yield accumulated grammatical-acoustic scores for whole phrases.

Quite the opposite problem arises in the analysis context when we ask for the (relative) probability of a given string y being derived by a particular derivation T_k (when there may be several different derivation histories T_i for the same string).

We may compute the a-posteriori derivation probability of a string y by using Bayes' Theorem

$$p(S \leftarrow T_k \leftarrow y) = \frac{p(y \leftarrow T_k \leftarrow S)}{\sum_i p(y \leftarrow T_i \leftarrow S)}$$

As a specialization, this formula is of particular interest if we want to predict e.g. words or categories following or preceding a already recognized word etc. (This is useful for 'island parsing' when only the most promising parses should be continued.)

Consequently, the a-posteriori probability that the lexical category Q immediately follows the word 'a' can be calculated as

$$p(S \leftarrow x a Q y) = \frac{\sum_i p(x_i a Q y_i \leftarrow T_i \leftarrow S)}{\sum_j p(w_j a P_j z_j \leftarrow T_j \leftarrow S)}$$

All derivations appearing on the right side are minimal derivations for the substring "aQ" or "aP_j" and the P_j's range over all lexical categories in G (In the formula, of course, we assume p(waPz <- S) = 0 if the substring "aP" isn't derivable in G). This formula reflects the common probabilistic assumption that the derivation probability of a substring is the sum of all distinct alternative derivation probabilities of this string (if there is more than one possibility).

2.3. Example Grammar G1

The following toy grammar is designed to demonstrate the formalism. That it generates many unwanted sentences need not concern us here.

Our grammar has the following rules

S	->	# NP V NP #	,	1.0
NP	->	Q N	,	0.7
NP	->	Q	,	0.3

Lexical rules

N->	board	0.2	V->	board	0.3
N->	boards	0.2	V->	boards	0.3
N->	men	0.3	V->	boarded	0.3
N->	man	0.3	V->	man	0.1
Q->	some	0.4	Q->	the	0.6

Let us assume the word "board" has been recognized somewhere in the input stream (but not at its end). We obtain the following a-priori probabilities for minimal derivations involving "board" with a subsequent lexical category

$$\begin{aligned} p(\# Q \text{ board } V \text{ NP } \# \leftarrow S) &= 0.7 * 0.2 \\ p(\# NP \text{ board } Q \text{ N } \# \leftarrow S) &= 0.3 * 0.7 \\ p(\# NP \text{ board } Q \# \leftarrow S) &= 0.3 * 0.3 \end{aligned}$$

Actually, there are no more minimal derivations of the desired type. We may now calculate the a-posteriori probability of V following the word "board"

$$p(\# x \text{ board } V y \# \leftarrow S) =$$

$$\frac{0.7 * 0.2}{0.7 * 0.2 + 0.3 * 0.7 + 0.3 * 0.3} = 0.32$$

The a-posteriori probability of the other ("conflicting") possibility i.e. that a Q follows the word "board" is

$$p(\# \text{ x board Q y } \# \text{ } < \text{ } S) = 1 - 0.32 = 0.68$$

In our implementation these a-posteriori probabilities can easily be computed from the derivation probabilities attached to the active edges in the chart parser.

3. Lexical Functional Grammar

LFG assumes two layers of grammatical description of sentences i.e. the constituent structure level and the functional structure level. The constituent structure level caters for the surface oriented realization of sentences (e.g. word order etc.) whereas the functional structure level is concerned with more abstract and supposedly universal grammatical functions like SUBJECT, OBJECT, OBLIQUE object and the like. Lexical functional Grammars use context-free rules (like in the example above) coupled with functional schemata. These schemata (normally) relate F-structures associated with corresponding mother and daughter nodes in a c-structure (roughly speaking). The functional schemata attached to lexical items so-called semantic forms may include grammatical or semantic features, but more important, they allow a case frame notation (in particular important with verbs). It is these case frames (or valencies) that make LFG in particular attractive for prediction purposes in speech recognition.

In the implementation of the LFG system F-structures are incrementally constructed by using *unification*, i.e. a process that accumulates information in structures and never backtracks. This process is independent of the particular order in which these structures are constructed - an important aspect in speech recognition where there is inherently no predetermined order of the operations to follow.

3.1. Example Grammar G2

The following small grammar fragment should give a rough impression of the basic features of our approach. Trivial rules are omitted. Since we work within a railway inquiry environment we take special care of locative and temporal expressions. As an example, we have a special lexical category for place and station names (N-loc) and for time intervals like "day" and "week" etc (N-temp). A particular problem in LFG is the treatment of (oblique) objects and free adjuncts. In our context, we assume all temporal modifiers to be free adjuncts and verbs to be subcategorizable for oblique locative objects only (besides the normal arguments SUBJ, OBJ etc.). Our approach differs from /Bresnan 1982/ in various aspects. (Technically speaking, functional schemata of the

form ($\uparrow$ ($\downarrow$...)) = $\downarrow$ pose certain problems for structure prediction (generation). So we avoid them.

S $\rightarrow$ {AUX} NP VP { PP-temp }
 $(\uparrow \text{ SUBJ }) = \downarrow$ $(\uparrow \text{ ADJUNCT }) = \downarrow$

Temporal propositional phrases are treated as adjuncts.

S $\rightarrow$ XP AUX S
 $\left\{ \begin{array}{l} \uparrow \text{ OBJ} = \downarrow \\ \uparrow \text{ OBLLOC} = \downarrow \end{array} \right\}$

This is the rule for questions with a question element in front.

VP $\rightarrow$ V { NP }
 $(\uparrow \text{ OBJ }) = \downarrow$

VP $\rightarrow$ V { PP-loc }
 $(\uparrow \text{ OBLLOC }) = \downarrow$

Verbs take a direct or oblique locative object.

PP-loc $\rightarrow$ P NP
 $(\uparrow \text{ OBJ }) = \downarrow$

Lexicon

call V $(\uparrow \text{ PRED }) = \text{"CALL"} < (\uparrow \text{ SUBJ }) (\uparrow \text{ OBLLOC }) >$
 $(\uparrow \text{ OBLLOC OBJ PCASE }) = \text{Loc}$

This lexical rule is viewed in the bottom-up analysis process as predicting a subject and an oblique object to appear somewhere in the sentence.

depart V $(\uparrow \text{ PRED }) = \text{"DEPART"} < (\uparrow \text{ SUBJ }) (\uparrow \text{ OBLLOC }) >$
 $(\uparrow \text{ OBLLOC OBJ PCASE }) = \text{Goal}$

This entry predicts a subject and an oblique object which denotes a goal (like in "depart ...for..." or "depart...to...").

arrive V $(\uparrow \text{ PRED }) = \text{"ARRIVE"} < (\uparrow \text{ SUBJ }) (\uparrow \text{ OBLLOC }) >$
 $(\uparrow \text{ OBLLOC OBJ PCASE }) = \text{Source}$

at P-loc $(\uparrow \text{ PRED }) = \text{"AT"} < (\uparrow \text{ OBJ }) >$
 $(\uparrow \text{ OBJ PCASE }) = \text{Loc}$

to P-loc $(\uparrow \text{ PRED }) = \text{"TO"} < (\uparrow \text{ OBJ }) >$
 $(\uparrow \text{ OBJ PCASE }) = \text{Goal}$

for P-loc $(\uparrow \text{ PRED }) = \text{"FOR"} < (\uparrow \text{ OBJ }) >$
 $(\uparrow \text{ OBJ PCASE }) = \text{Goal}$

5. References

- Block, H-U., Hunze, R.,
"Incremental Construction of C- and F-
Structures in a LFG-Parser",
Proc. COLING-86, Bonn, 1986, p. 490 - 493
- Bresnan, J. (ed) ,
The Mental Representation of
Grammatical Relations,
MIT Press 1982
- Ermann, L., Hayes-Roth, F., Lesser, V., Ray Reedy,
"The Hearsay-II Speech Understanding
System: Integrating Knowledge to
Resolve Uncertainty",
Computing Surveys, v.12, n.2, June 1980
- Levelt, W.,
Formal Grammars in Linguistics and
Psycholinguistics,
Vol 1, Mouton 1974
- Winograd, T.,
Language as a Cognitive Process,
Addison-Wesley, 1983