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 IKAROS[1] *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. Grammaticaless 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 $G = < V_N, V_T, R, S >$:

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 $\{R_1, R_2,\ldots, R_i\}$
with each rule having the format
$R_i = < A_i \rightarrow X_i , q_i >$
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

$$L_j \quad = < A_j -> t_j , q_j >$$

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 :

$$p(S \quad <- \quad S) = 1$$

$$p(xYz \quad <-T- \; S) = \quad p(xAy <- S) * q \quad ,$$
$$\text{if there is a rule } < A -> Y , q> \text{ in } R$$

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

$$p (S <- T_k - \; y) = \frac{p(\; y \; <- T_k - \; S)}{\sum_i p(\; y \; <- T_i - \; S)}$$

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

$$p(S \; <- \; x\, a\, Q\, y \;) = \frac{\sum_i p(\; x_i \; a \; Q \; y_i \; <-T_i- \; S \;)}{\sum_j p(\; wj \; a \; Pj \; zj \; <-Tj- \; S \;)}$$

All derivations appearing on the right side are minimal derivations for the substring "aQ" or "aPj" and the Pj´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

p(# Q board V NP # <- S) = 0.7 * 0.2
p(# NP board Q N # <- S) = 0.3 * 0.7
p(# NP board Q # <- S) = 0.3 * 0.3

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 board V y # <- 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(# x board Q y #   <-   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 fuctional 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 intervalls 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 (↑ (↓...)) = ↓ pose certain problems for structure prediction (generation). So we avoid them.

```
S      ->{AUX}    NP                VP   { PP-temp}
                  (↑ SUBJ ) = ↓          (↑ADJUNCT)=↓
```

Temporal propositional phrases are treated as adjuncts.

```
S      ->           XP              AUX         S
              {   ↑OBJ = ↓    }
              {↑OBLLOC = ↓}
```

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

```
VP    ->    V     { NP }
                  (↑ OBJ ) = ↓

VP    ->    V     { PP-loc }
                  (↑  OBLLOC) = ↓
```

Verbs take a direct or oblique locative object.

```
PP-loc->    P     NP
                  (↑ OBJ ) = ↓
```

Lexicon

```
call   V    (↑PRED)="CALL<(↑SUBJ) (↑OBLLOC)>"
            (↑OBLLOC OBJ  PCASE)      = 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   (↑PRED)="DEPART<(↑SUBJ)(↑OBLLOC)>"
            (↑OBLLOC OBJ  PCASE)      = Goal
```

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

```
arrive  V   (↑PRED)="ARRIVE<(↑SUBL)(↑OBLLOC)>"
            (↑OBLLOC OBJ  PCASE)      = Source

at          P-loc     (↑PRED )   = "AT<(↑OBJ )>"
                      (↑ OBJ  PCASE ) = Loc

to          P-loc     (↑PRED )   = "TO<(↑OBJ )>"
                      (↑ OBJ  PCASE ) = Goal

for         P-loc     (↑PRED )   = "FOR<(↑OBJ )>"
                      (↑ OBJ  PCASE ) = Goal
```

$$\begin{array}{lll} \text{where} & \text{XP} & (\uparrow \text{PRED}) = \text{"WHERE"} \\ & & (\uparrow \text{OBLLOC OBJ PCASE}) = \begin{Bmatrix} \text{Loc} \\ \text{Goal} \end{Bmatrix} \end{array}$$

This rule reflects the fact that "where" may play the role of an oblique location or goal object (like in examples "Where does the train stop" and "Where does the train go" but not in "From where does the train arrive").

$$\begin{array}{lll} \text{Coventry} & \text{N-loc} & (\uparrow \text{PRED}) = \text{"COVENTRY"} \end{array}$$

This is an example entry for a place name.

$$\begin{array}{lll} \text{day} & \text{N-temp} & (\uparrow \text{PRED}) = \text{"DAY"} \end{array}$$

For the analysis of the sentence "where did the train call" we get the c-structure

$[_S [_{XP}\text{where}] [_{AUX}\text{ did}][_S[_{NP}\text{the train}] [_{VP}[_V\text{ call}]]]]$

and the f-structure

$$\begin{array}{llll} \text{SUBJ} & = & \text{the train} & \\ \text{PRED} & = & \text{"CALL<}(\uparrow \text{SUBJ})(\uparrow \text{OBLLOC})\text{>"} & \\ \text{OBLLOC} & \text{PRED} & = & \text{"WHERE"} \\ & \text{OBJ PCASE} & = & \text{Loc} \end{array}$$

In order to demonstrate the hole-filling capabilities of this formalism we consider the phrase "call * Coventry" with * indicating a word that was not recognized by the acoustic-phonetic component. We would get the c-structure

$[_{VP} [_V\text{ call} [_{PP\text{-}loc} [_{P\text{-}loc} *] [_{N\text{-}loc}\text{ Coventry}]]]]$

and the f-structure

$$\begin{array}{llll} \text{PRED} & = \text{"CALL<}(\uparrow \text{SUBJ})(\uparrow \text{OBLLOC})\text{>"} & & \\ \text{OBLLOC} & \text{PRED} & = & \text{"Coventry"} \\ & \text{OBJ PCASE} & = & \text{Loc} \end{array}$$

This little example shows how our LFG-approach is capable to predict certain features of constituents that might appear somewhere in the sentence.

Now, another important point is that *LFG subcategorizes for grammatical functions not for grammatical categories.* That means we have a certain flexibility at hand in that the same grammatical function (e.g. the Location deep case) may be realized in different ways (compare for instance the example sentence in L(G2) "Where did the train call" with a WH-Adverb vs. "The train calls at Coventry" with an oblique object). As the example clearly shows, grammatical functions in LFG provide an additional intermediate level of description between a semantic feature approach("semantic grammars") and a purely surface oriented word order approach.

Since there are sentences that are syntactically quite acceptable (i.e. on the constituent structure level) but devious in semantic terms LFG imposes 3 additional well-formedness conditions on F-structures. We have to assess these conditions from the pragmatic viewpoint of a real-life application (e.g. with respect to predictive power and error tolerance)

(i) Functional Uniqueness (no conflicting values for an attribute allowed)
This is a useful principle since we want to exclude feature "clashes". So we would like to exclude "Where did the train stops" (tense clash) but we would not want to undertake great an effort in order to exclude "Where does the train stops" ("since it is clear what is meant!").

(ii) Completeness (A f-structure must contain all the governable grammatical functions that its predicate governs)
This is an awkward condition. First of all, given the uncertainty in speech recognition it is hard to decide at any rate when the analysis of several (conflicting) utterances is complete. In addition, we believe that there are never ending problems with the distinction between obligatory and optional arguments of a verb. Hence we decided that all arguments in a semantic form should be regarded as optional (Only SUBJ is obligatory). A f-structure that contains more grammatical functions (out of the list given in the predicate) is grammatical better than one with less functions in itself.

(iii) Coherence (There must be no grammatical function in a f-structure that is not governed by a predicate)
This is a good principle since we want to exclude superfluous arguments.

4. Conclusions

We showed the usefulness of a probabilistic lexical functional grammar for a speech recognition system by demonstrating its two relatively independent constraining and predicting mechanisms : the constraining power of a context-free grammar (which allows global predictions from a global point of view) and of valency-oriented lexicon (which allows bottom-up predictions from a local point of view). In addition, we gave an account of the probability treatment within this framework.

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