AN AUGMENTED CHART PARSING ALGORITHM INTEGRATING UNIFICATION GRAMMAR AND MARKOV LANGUAGE MODEL FOR CONTINUOUS SPEECH RECOGNITION

Lee-feng Chien*, K. J. Chen** and Lin-shan Lee*

* Dept. of Computer Science and Information Engineering, National Taiwan University,
Taipei, Taiwan, Rep. of China, Tel: (02) 362-2444.
** The Institute of Information Science, Academia Sinica, Taipei, Taiwan, Rep. of China

ABSTRACT

In this paper, an efficient algorithm is developed to handle the difficulties in parsing noisy word lattices (sets of word hypotheses obtained in continuous speech recognition) which include problems such as word boundary overlapping, homonyms, lexical ambiguities, recognition uncertainty and errors, etc. An augemented chart is first proposed, and the new algorithm is then derived on this chart. This algorithm properly integrates the global structural synthesis capabilities of the unification grammar and the local relation estimation capabilities of the Markov language model. The parsing algorithm is island-driven and best-first. In this way, not only the features of the grammatical and statistical approaches can be combined, but the effects of the two different approaches are reflected in a single algorithm such that the overall selectivity can be appropriately optimized.

1. INTRODUCTION

In this paper, an efficient parsing algorithm integrating unification grammar and Markov language model is developed to handle the difficulties in parsing noisy word lattice (sets of word hypotheses obtained in continuous speech recognition) which include problems such as word boundary overlapping, homonyms, lexical ambiguities, recognition uncertainty and errors, etc. These are the problems encountered in parsing spoken natural language, usually not present in parsing typed natural language. It is therefore not possible to directly apply techniques for parsing typed natural language to spoken natural language [1]. Several algorithms for parsing word lattices had been proposed [2,3] and shown to be very efficient in parsing less ambiguous natural languages such as English obtained in speech recognition. However, all of them are primarily strictly left-to-right, thus with relatively limited applications for cases in which other strategies such as island-driven [1] are more useful, for example, corrupted word lattice with extra, missing or erroneous phones in speech recognition [4].

Since the most important problem in all speech recognition systems is the inherent uncertainty associated with the acoustic-phonetic processing in such systems. Language processing for speech recognition should therefore be able to tolerate errors of recognition, and even to compensate for some errors occurring in the acoustic-phonetic phase. Conventionally there are two approaches of language modeling: grammatical and statistical. The features of these two approaches are basically complementary -- the statistical approaches predict very well locally, and can even tolerate some errors of recognition; while the grammatical approaches are better in checking the global structures, finding out the exact meaning or sentence structure, and can sometimes compensate for recognition errors. A unified approach integrating the grammatical and the statistical approaches is thus desired to combine the advantages of both approaches to improve the correct rate of recognition [5]. In this paper, a new parsing algorithm is proposed to integrate unification grammar and Markov language model for continuous speech recognition applications. Unification-based grammars (UG's), e.g. LFG, GPSG, have some advantages in language processing, such as surface-based, informational, inductive, declarative, ..., etc. [6]. It is also easy to integrate syntactic information (e.g. categories, subcategorization) with semantic information (e.g. case roles, semantic markers) by UG's and such advantages had been applied to speech recognition [3]. Markov language model [7], on the other hand, has the advantages that the model parameters can be statistically trained, the results are easy to determine, and the acoustic recognition uncertainty can be included. Therefore two very promising approaches are integrated together in this new algorithm.

The proposed algorithm is based on an augmented chart. Chart has been an efficient working structure widely used in many natural language processing systems [8] but it is basically designed to parse a sequence of fixed and known words instead of ambiguous word lattice. In this paper, the conventional chart is extended or augmented such that it is able to represent a word lattice; and the new word lattice parsing algorithm is then derived on this chart. This algorithm properly integrates the global structural synthesis capabilities of the unification grammar and the local relation estimation capabilities of the Markov language model. The parsing algorithm is island-driven and best-first, and the augmented chart representation can carefully consider the complicated problems in noisy word lattices. In this way, not only the features of the grammatical and statistical approaches can be combined, but the effects of the two different approaches are reflected and integrated in a single algorithm such that the overall selectivity can be appropriately optimized.

In the following, Section 2 formally describe the problem, the unification grammar and the Markov model. Section 3 introduce the concept of the augmented chart and the procedure to map an input word lattice to the augmented chart. The new augmented chart parsing algorithm integrating the two approaches is then presented in Section 4; while some preliminary experimental results and concluding remarks are finally given in Section 5.

2. THE PROBLEM DEFINITION AND LANGUAGE MODELING

Some relevant definitions are first given below.

Word Lattice: A word lattice W is a partially ordered set of word hypotheses, $W = \{w_1, ..., w_m\}$, where each word hypothesis w_i, $i=1,...,m$, is characterized by *begin*, the beginning point, *end*, the ending point, *cat*, the category, *fea*, the feature structure, *phone*, the associated phonemes, and *name*, the word name of the word hypothesis. These word hypotheses are sorted in the order of their ending points; that is, for every pair of word hypotheses w_i and w_j, $i<j$ implies $\mathrm{end}(w_i) <= \mathrm{end}(w_j)$. Also, two word hypotheses w_i and w_j are said to be connected if there is no other word hypothesis located exactly between the boundaries of the two word hypotheses, i.e., if $w_i \leq w_j$ and there does not exist any other word hypothesis w_k such that $w_i \leq w_k \leq w_j$, where $w_i \leq w_j$ iff $\mathrm{end}(w_i) <= \mathrm{begin}(w_j)$. A sentence hypothesis $S = \{w_{i1}, w_{i2}, ..., w_{in}\}$ is then a sequence of connected word hypotheses from a starting word to an ending word selected from the given word

CH2847-2/90/0000-0585 $1.00 © 1990 IEEE

lattice, and a sentence hypothesis is grammatically valid only if it can be generated by a grammar. As an example, a sample word lattice constructed for demonstration purpose is shown on the top of Fig. 1, in which only the word sequence "Tad does this." is a grammatically valid sentence hypothesis.

Unfication Grammar (UG): The unification grammar used here is a PART-II-like formalism [6]. It is composed by a lexicon and a set of combinatory rules. Every word in the lexicon is represented as a kind of feature structures. Combinatory rules are employed to describe how strings are concatenated to form longer strings and how the associated feature structures are related. A UG is a four-tuple (V_n, V_t, S, R), where V_n is a set of phrase symbols, such as NP, VP, ...etc.; V_t is a set of category symbols, such as N, V, Det, ...etc.; S is the sentence symbol belonging to V_n; R is a set of combinatory rules, where each rule r is defined as r= (α_i --> β_{ij}, E_{ij}), i=1,...,k, j=1, ...,n_i ,where $\alpha_i \in V_n$, $\beta_{ij} \in (V_n \cup V_t)^*$ and E_{ij} is a set of constraint equations describing relations among the elements of the associated feature structures of α_i and β_{ij}. Unification is then prescribed as the sole information-combining operation. This makes the formation completely declarative and its interpretation order-independent. The unification operation in our formalism is described by the constraint equations. In the following, some example rules of the UG are illustrated, where below each PS rule is some associated constraint equations.

```
(r1)  S --> NP     VP            (r3)  VP --> V,
      <VP Subj> =  <NP>                <VP> =  <V>
      <S> = <VP>                 (r4)  VP --> V  NP,
(r2)  NP --> N                         <V Obj> =  <NP>
      <NP> =  <N>                      <VP> =  <V>
```

Markov Language Model: A direct relation to compute the probablitiy P(S) for a sentence hypothesis S = {w_1, w_2, ..., w_n} is

$$P(S) = \prod_{i=1}^{n} P(w_i|w_{i-1}, w_{i-2},...w_1)$$

but practically such a relation will cause difficulties in implementation because even for a moderate vocabulary size the number of the probabilities $P(w_i|w_{i-1}\ w_{i-2},...w_1)$ would be too large to estimate, store or retrieve. In this parsing algorithm we simplify the above relation by assuming a first-order Markov language model (bi-gram model), i.e.,

$$P(w_i|w_{i-1},w_{i-2},w_1) = P(w_i|w_{i-1})$$

and the probabilities $P(w_i|w_{i-1})$ are estimated using a large database of training corpus. These probabilities are estimated by simply counting the frequencies of occurrence for different words to appear in adjacent positions.

We now formally define the problem addressed in this paper below.

The Problem: Given a word lattice W = {w_1, ... , w_n}, a unification grammar G and a Markov langauage model M, the purpose of the parsing algorithm is to efficiently and accurately find out a sentence hypothesis S^* that is grammatically valid with a satisfactory probability $P(S^*)$ based on W, G and M.

Assume a sentence hypothesis is denoted by S = w_1,w_2, w_3.....w_n. Let U denote the unknown input speech signal. In statistical point of view, a very natural decision rule for a speech recognition system to decide in favor of a sentence hypothesis S^+ is

$$P(S^+ \mid U) = \max_{S} P(S \mid U)$$

in other words, the sentence hypothesis S^+ is chosen if it maximizes the probability P(S |U) for all possible sentence hypotheses S. In this paper, in order to integrate the statistical and grammatical approaches, the decision rule is modified such that the purpose is to find a sentence hypothesis S^* that is both grammatically valid and with a satisfactory probability $P(S^*|U)$, then the component words and the associated feature structure of S^* are the recognized result. Here we replace the highest probability by a satisfactory one because exhaustively parsing a word lattice is often computational inefficient. Using the Bayes formula,

$$P(S \mid U) = \frac{P(U \mid S)\, P(S)}{P(U)}$$

For a given unkonwn speech signal U, P(U) is the same for all possible sentence hypotheses S. Therefore all the speech recognition system has to do is to find a sentence hypothesis S which is grammatically valid and P(U|S) P(S) is satisfactory. Since the first term P(U|S) can be obtained in the acoustic-phonetic processing phase using approaches such as hidden Markov models, in this paper we will focus on the evaluation of P(S) and the grammatical analysis of S.

In the new parsing algorithm proposed in this paper, the grammatical analysis is based on UG and the probability estimation is based on the Markov language model. In parsing, each constituent created according to the description of UG will be assigned a probability based on the Markov language model, in which the probability for the component word string of the constituent is taken as the probability of the constituent. Therefore the constituent with the highest probability will be constructed first onto the augmented chart. The first sentence hypothesis constructed from a starting word to an ending word with a satisfactory probability is the recognized result. If the input lattice is seriously corrupted, there may be no grammatically valid sentence hypothesis existing in the word lattice. With this algorithm because all partial parsed results are all recorded in the augmented chart, thereby some compensatory strategies, such as to continue to find the word string $S^\#$ that maximizes P(S|U) or an island (constituent) which covers the largest number of words, can be selected to give reasonable results. Since all of the operations are performed on the augmented chart, in the following sections, we will first describe the augmented chart and then the algorithm.

3. THE AUGMENTED CHART

The conventional chart parsing algorithm was designed to parse a sequence of words. Here the chart is augmented for parsing word lattices. The augmented chart is a directed uncyclic graph specified by a two-tuple <V, E>, where V is a sequence of vertices and E is a set of edges. Each vertex in V represents an end point of some word hypotheses in the input word lattice, while the edge set is divided into three disjoint groups: inactive, active and jump edges. As were used in a conventional chart, an inactive edge is a data structure to represent a completed constituent. It is characterized by the following information: *from*, the vertex where the edge starts (the begin vertex), *to* , the vertex where the edge ends (the end vertex), *cat,* the associated category, *P*, the associated probability, *sub-inactive,* the list of the immediately spanned inactive edges that were included and *name,* the word name (for lexical edges only). An inactive edge is called a lexical edge if it has a lexical category, otherwise it is a phrasal edge. An active edge represents an incomplete constituent which needs some other complete constituents to compose a larger one. It is similarly characterized as above by *from, to, P, sub-inactive*, as well as *rule,* the referred grammar rule and *pos,* the position of the category in the rule it is looking for. A jump edge, however, is a functional edge which links two different edges to indicate their connection relation (described below) and guide the parser to search through all edges connected to each active edge during parsing. The partial ordering relation among the edges in the augmented chart can first be defined according to the order of the boundary vertices. Two edge E_i and E_j are then said to be connected (i.e. EConn(E_i, E_j) = true) only when the end vertex of one of them is the begin vertex of the other, or there exists a jump edge linking them together. For example, in the chart representation of the sample word lattice in Fig. 1 (on the bottom of the figure, the details will be explained in the following), EConn(E_3, E_6) = true due to the existence of Jump3 linking E_3 and E_6, but EConn(E_1, E_6) = false due to E_3 and E_4 existing in between. This jump edge and the new connection relation is the primary difference between the conventional chart and our augmented chart.

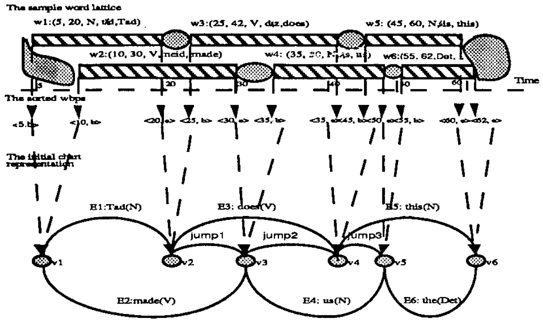

Fig.1 In this figure, on the top is a set of overlapped word hypotheses which are assumed to be produced by an acoustic signal processor in speech recognition, where each rectangular shape denotes the time segment of the acoustic signal for the word hypothesis and above it is the 6-tuple information, from left to right, i.e., *begin*, *end*, *cat*, *phone* and *name*, respectively (*fea* has been left out for simplicity); on the middle are the sorted wbp's; and on the bottom is the resulting initial chart.

Before parsing, any input word lattice has to be mapped to the augmented chart such a procedure is described below. At the beginning of the mapping procedure, we have to first consider a situation in which additional word hypotheses should be inserted into the input lattice to avoid any important word being missed in the sentence. A good example for such a situation is in Fig. 2 where the time segment for the word hypothesis w_i (the word "same") is from 10 to 20, and that for w_j (the word "message") is from 14 to 30. Apparently for this situation four cases are all possible: w_i is a correct word but w_j is not, w_j is correct but w_i is not, both w_i and w_j are correct because they share a common phoneme (m) in the co-articulated continuous acoustic signal, or both w_i and w_j are not correct. A simple approach to be used here is that two additional word hypotheses w_{i1} (also "same", but from 10 to 17) and w_{j1} (also "message", but from 17 to 30) are inserted into the word lattice W, such that all the above four possible cases will be properly considered during parsing and no any word will be missed.

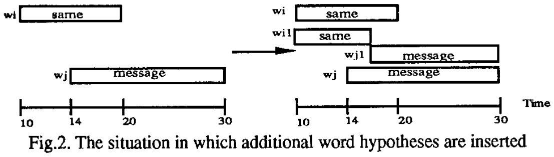

Fig.2. The situation in which additional word hypotheses are inserted

After the above additional word hypotheses insertion, every boundary point (either beginning or ending) of any word hypothesis of W should then be mapped to a vertex in the chart. All these word boundary points (wbp's) have to be first sorted into an ordered sequence (indicated by a function Order(x), where x is any wbp); the definition of Order(x) is as follows. To any pair of wbp's x and y, if x and y are distinct then their order is based on order in time; if x and y are identical then the beginning wbp (denoted by *b*) is after the ending wbp (denoted by *e*). For each wbp x, the corresponding vertex is then assigned depending on its preceding wbp y as described below. As was shown in Fig. 3, for totally four possible cases of x and y, i.e. *bb* (y is a beginning wbp and x is also a beginning wbp), *be*, *eb*, *ee*, only for the case *be* (y is a beginning wbp but x an ending wbp), two different vertices should be assigned to x and y to preserve the ordering relation between the corresponding word hypotheses of x and y. But in all the other three cases, x and y can be given the same vertex. Let the function Vertex(x) denotes this assignment.

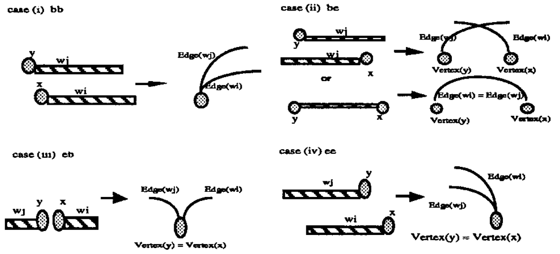

Fig. 3. Vertex assignment of the word boundary points

Now, for each word hypothesis w_i , an initial inactive edge can be constructed. The function Edge(w_i) for a word hypothesis w_i is then exactly specified by the two vertices assigned to the two wbp's of w_i , i.e. Edge(w_i) = < Vertex(begin(w_i)), Vertex(end(w_i))>. Finally, for any pair of vertices v_i and v_j, if there isn't any complete initial inactive edge existing between them, a jump edge from v_i to v_j is constructed to link v_i and v_j. Using the above procedure, Fig. 1 also shows the mapping results of the sample word lattice. The sorted wbp's (specified by a time scale and whether it is a beginning or ending wbp) are on the middle of the figure, and the resulting initial chart is on the bottom. It can be shown that the above mapping procedure has the following nice properties: first, the ordering and connection relations among all word hypotheses in the word lattice can be completely preserved among the corresponding edges in the augmented chart; second, when the input word lattice can be reduced to a simple sequence of word hypotheses, the augmented chart representation can also be reduced to a conventional chart representation.

4. THE AUGMENTED CHART PARSING ALGORITHM

The fundamental principle of chart parsing is: Whenever an active edge A is connected to an inactive edge I which satisfies A's conditions for extensions, a new edge N covering both is built. Now, in the augmented chart parsing this principle is still held; except that the inactive edge I doesn't have to share the same vertex with the active edge A; instead it can be separated from the active edge A, as long as there exists a jump edge linking edges A and I. Meanwhile, the conditions to be met should include some additional information such as all associated unification operations being successful, etc. Moreover, for each constituent C, a probability P(C) = P(W_c) is assigned, where W_c is the component word hypothesis sequence of C and P(W_c) is obtained from the Markov language model. When an active constituent A and inactive constituent I form a new constituent N, the probability P(N) can be easily evaluated from probabilities P(A) and P(I), i.e. $P(N) = P(A)*P(I)*\{P(w_{i1}|w_{am})/P(w_{i1})\}$, where w_{i1} is the first word hypothesis of I and W_{am} is the last word hypothesis of A if A is to the left of I. This is explained below. Let W_n, W_a, W_i be the component word hypothesis sequences of N, A, and I respectively. Without loss of generality, we assume A is to the left of I, thereby $Wn = WaWi = w_{a1},...,w_{am},w_{i1},...,w_{in}$. Then, $P(W_n) = P(WaWi)$

$$=P(w_{a1}) *\underset{2\le k\le m}{\pi} P(w_{ak}|w_{ak-1})*P(w_{i1}|w_{am})*\underset{2\le k\le n}{\pi} P(w_{ik}|w_{ik-1})$$

$$= P(W_a)*P(W_i)*\{P(w_{i1}|w_{am})/P(w_{i1})\}.$$

This can be easily evaluated in each parsing step. Although P(C) is assigned to every constituent in the augmented chart, only the constituents with the highest probabilities P(C) will be constructed. The first sentence constituent constructed from the starting word to the ending word with a satisfactory probability is the recognized result. In fact the above concepts form a useful scheme that can be extended to develop different parsing algorithms on the augmented chart for different speech recognition applications. In this paper, a bottom-up and unidirectional island-driven (searching actions triggered by an island are always from left to right) parsing algorithm is illustrated.

This algorithm consists of a main procedure: Parser, with five assistant procedures: Chart-Initialization, Expectation -Formation, Edge-Construction, Pop-Agenda and Push-Agenda; and two global data structures: chart and agenda. An abstract diagram depicted in Fig. 4 clearly indicates the relations among the five assitant procedures and the two global data structures.

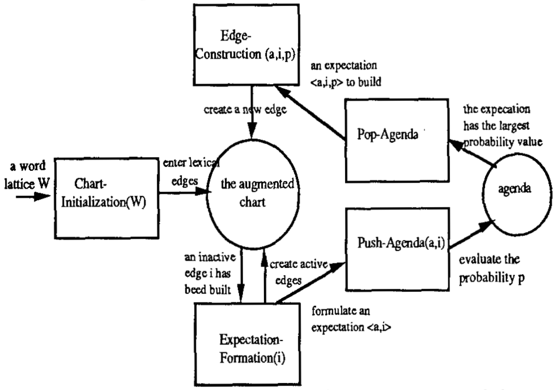

Fig. 4 The relations among the five assistant procedures and the two global data structures

Initially, a word lattice is given to the procedure Parser, and every word hypothesis of it is represented as a lexical inactive edge on the augmented chart by calling the assistant procedure Chart-Initialization. Then, for each of the lexical edges, the procedure Parser will continuously call the assistant procedure Expectation-Formation to establish all probable expectations. An expectation is a set of information <a, i, p> to denote the possible combination of the active edge E_a, the inactive edge E_i, as well as the probability p of the combination. Therefore the function of the assistant procedure Expectation-Formation is, for any given inactive edge E_i, if $cat(E_i)$ is the first category symbol at the right hand side of some rules then to each of them create an active edge to denote a searching request; or if there exist some active edges and each of them, namely E_a, is left connected to E_i and $cat(E_a) = cat(E_i)$ then to each E_a formulates an expectation and call the assistant procedure Push-Agenda to insert the expectation to agenda. The agenda is an ordered list of expectations which are ready to be constructed onto the chart. Also the assistant procedure Push-Agenda will evaluate the probability p for each given expectation by means of the previously described evaluation method, and the expectation <a, i, p> in agenda with the highest probability p will be popped from agenda by calling the assistant procedure Pop-Agenda and passed to the assistant procedure Edge-Construction to build a new edge from the expectation onto the chart. If $rule(E_a)$ is satisfied and the unification operations which $rule(E_a)$ describes are all successful then an inactive edge is built; or if there are some other necessary categories in $rule(E_a)$ then an extended active edge is built. When the first grammatical sentence constituent formed from a starting word hypothesis to an ending word hypothesis has been constructed with a satisfactory probability, the parsing process is completed. While if the agenda is empty and no any satisfactory sentence hypothesis is found, some compensatory routines can be used to process the corrupted lattice.

5. PRELIMINARY EXPERIMENTAL RESULTS AND CONCLUDING REMARKS

In order to see how the above concept works, a bottom-up parser based on the proposed parsing algorithm was developed and tested on a small word dictionary with about 1500 words, a Markov language model, and a simple set of unification grammar rules for Chinese language. A large set of Chinese word lattices obtained by an acoustic signal processor which recognizes Mandarin speech is used as the input to the parser. Due to large number of homonyms existing in Chinese language and uncertainty and errors in speech recognition, very high degree of lexical ambiguity exists in the input lattices. The preliminary results indicate that the augmented chart representation can carefully consider the complicated problems in noisy word lattices, and the joint effect of the global syntactic and semantic analysis capabilities of the unification grammar and the local relation estimation capabilities of the Markov language model can significantly reduce the interference of noisy word hypotheses. Significant improvements in computation complexity reduction and recognition accuracy were observed, although further experiments are still under progress.

REFERENCES:

[1] Hayes P. J. et al. (1986). Parsing Spoken Language: A Semantic Caseframe Approach. Proceedings of the International Conference on Computational Linguistics, pp. 587-592.

[2] Tomita M. (1986). An Efficient Word Lattice Parsing Algorithm for Continuous Speech Recognition. Proceedings of the International Conference on Acoustic, Speech and Signal Processing, pp. 1569-1572.

[3] Chow Yen-Lu and Ronkos Salim. (1989). Speech Understanding Using A Unification Grammar. Proceedings of the International Conference on Acoustic, Speech and Signal Processing, pp. 727-730.

[4] Ward W. H. et al. (1988). Parsing Spoken Phrases Despite Missing Words. Proceedings of the International Conference on Acoustic, Speech and Signal Processing, pp. 275-278.

[5] Derouault A. and Merialdo B. (1986). Natural Language Modeling for Phoneme-to-Text Transcription, IEEE Trans. on PAMI, Vol. PAMI-8.

[6] Sheiber S. M. (1986). An Introduction to Unification-Based Approaches to Grammar. University of Chicago Press, Chicago.

[7] Jelinek F. et al. (1983). Continuous Speech Recognition: Statistical Methods, IEEE Trans. PAMI., Vol. PAMI-5.

[8] Kay M. (1980). Algorithm Schemata and Data Structures in Syntactic Processing. Xerox Report CSL-80-12, Pala Alto.