An Augmented Chart Data Structure with Efficient Word Lattice Parsing Scheme In Speech Recognition Applications

Lee-Feng Chien*, K. J. Chen** and Lin-Shan Lee*

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

Abstract

In this paper, an augmented chart data structure with efficient word lattice parsing scheme in speech recognition applications is proposed. The augmented chart and the associated parsing algorithm can represent and parse very efficiently a lattice of word hypotheses produced in speech recognition with high degree of lexical ambiguity without changing the fundamental principles of chart parsing. Every word lattice can be mapped to the augmented chart with the ordering and connection relation among word hypotheses being well preserved in the augmented chart. A jump edge is defined to link edges representing word hypotheses physically separated but practically possible to be connected. Preliminary experimental results show that with the augmented chart parsing all possible constituents of the input word lattice can be constructed and no constituent needs to be built more than once. This will reduce the computation complexity significantly especially when serious lexical ambiguity exists in the input word lattice as in many speech recognition problems. This augmented chart parsing is thus a very useful and efficient approach to language processing problems in speech recognition applications.

1. Introduction

In this paper, the conventional chart data structure has been augmented for efficient word lattice parsing to handle the high degree of ambiguities encountered in speech recognition applications. A word lattice is a set of word hypotheses produced by some acoustic signal processor in continuous speech recognition applications which possibly includes problems such as word boundary overlapping, lexical ambiguities, missing or extra phones, recognition uncertainty and errors, etc. The purpose of parsing such a word lattice is to efficiently and accurately obtain the most promising candidate sentence at acceptable computation complexity by means of grammatical constraints and appropriate data structure design. For example, in the process of continuous speech recognition, it happened very often that not only more than one words may be produced for a given segment of speech (such as homonyms, especially for some languages with large number of homonyms such as Chinese language (Lee, 1987)), but many competing word hypotheses can be produced at overlapping, adjoining, or separate segments of the acoustic signal without a set of aligned word boundaries. This will result in huge number of sentence hypotheses, each of which formed by one combination of a sequence of word hypotheses, such that exhaustively parsing all these sentence hypotheses with a conventional text parser is computational inefficient or even prohibitively difficult. A really efficient approach is therefore desired. Several algorithms for parsing such word lattices had been proposed (Tomita, 1986;

Chow, 1989). These algorithms had been 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 from left-to-right, thus with relatively limited applications for cases in which other strategies such as island-driven (Hayes, 1986) or even right-to-left are more useful (Huang, 1988), for example, corrupted word lattice with extra, missing or erroneous phones in speech recognition (Ward, 1988). On the other hand, chart has been an efficient working structure widely used in many natural language processing systems and has been shown to be a very effective approach (Kay, 1980), 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 therefore extended or augmented such that it is able to represent a word lattice; while the conventional functions, operations and properties of a chart parser as well as some useful extensions such as the use of lexicalized grammars and island-driven parsing will not be affected by the augmentation at all. Therefore the augmented chart parsing proposed in this paper is a very efficient and attractive parsing scheme for many language processing problems in speech recognition applications. A word lattice parser based on the augmented chart data structure proposed here has been implemented and tested for Chinese language and the preliminary results are very encouraging.

In the following, Section 2 introduces the concept of the augmented chart and Section 3 describes the mapping procedure to map an input word lattice to the augmented chart. The parsing scheme and some further extensions are discussed in Sections 4; while some preliminary experimental results are presented in Section 5. Concluding remarks are finally given in Section 6.

2. The Augmented Chart

The conventional chart parsing algorithm was designed to parse a sequence of words. In this section the chart is augmented for parsing word lattices. The purpose is to efficiently and accurately find out all grammatically valid sentence hypotheses and their sentence structures from a given word lattice based on a grammar.

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, *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 $end(w_i) <= end(w_j)$. 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 $w_i \le w_j$ and there does not exist any other word hypothesis w_k such that $w_i \le w_k \le w_j$, where $w_i \le w_j$ iff $end(w_i) <= begin(w_j)$. A sentence hypothesis is then a sequence of connected word hypotheses selected from the given word lattice, and a sentence hypothesis is grammatical 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 valid sentence hypothesis.

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, while an active edge represents an incomplete constituent which needs some other complete constituents to compose a larger one. 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 next section), EConn(E_3, E_6) = true due to the existence of Jurnp3 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.

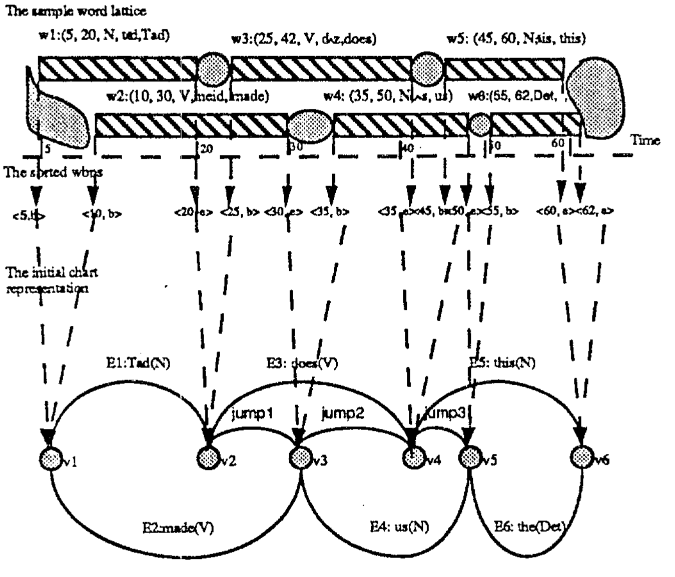

Fig.1 In this figure, on the top is a set of overlapped word hypotheses which are assumed to be produced by an acoustic signal processor in speech recognition, where each rectangular shape denotes the time segment of the acoustic signal for the word hypothesis and above it is the 5-tuple information, from left to right, i.e., ***begin***, ***end***, ***cat***, ***phone*** and ***name***, respectively; on the middle are the sorted wbp's; and on the bottom is the resulting initial chart.

3. The Mapping from a Word Lattice to the Augmented Chart

Before parsing is performed, any input word lattice has to be mapped to the augmented chart. 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 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.

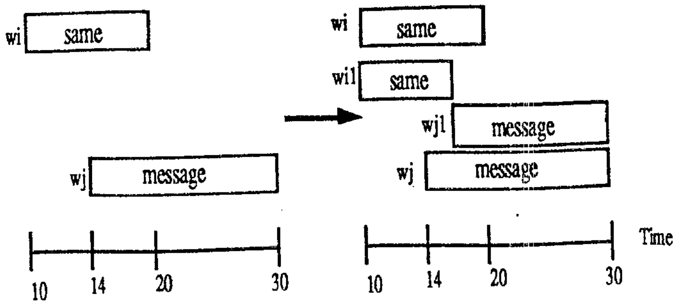

Fig. 2. The situation in which additional word hypotheses are inserted

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.

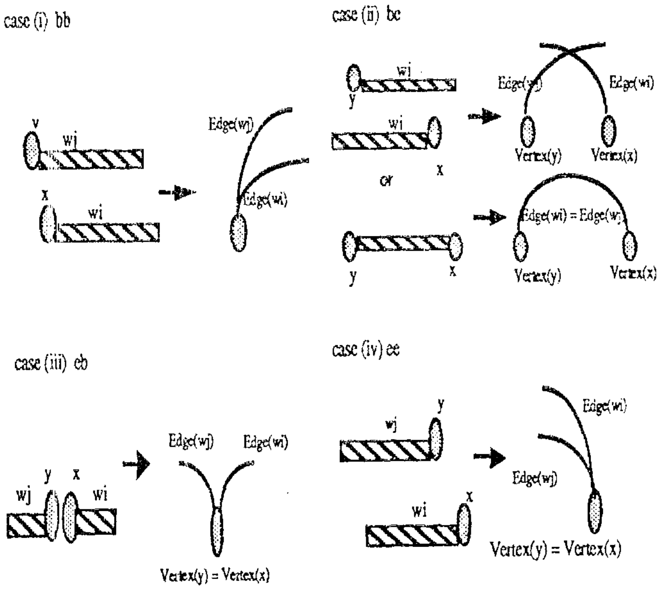

Fig. 3. Vertex assignment of the word boundary points

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 and Some Further Extensions

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. The augmented chart parsing scheme proposed here is not only very useful and efficient to rule-based grammar applications, but is equally useful and efficient in other applications such as a lexicalized grammar (e.g.

HPSG(Pollard, 1987)) in which the syntactical relationships are stated as part of the lexical description, and in the augmented chart the structures to be assigned to the input may be extended to attribute-value matrices (complex feature structures) instead of syntactic parsing trees and the recognition algorithm may rely on the head-driven slot and filler principle instead of derivation oriented recognition. Such an extension is in fact straightforward. Furthermore, in some other approaches to increase the flexibility of the slot and filler principle, such as island parsing (Stock, 1988) and discontinuous segmented parsing (Hellwig, 1988), the augmented chart proposed here can also be easily extended and applied.

5. Some Preliminary Experimental Results

In order to see how the above concept for augmented chart parsing works, a bottom-up and left-to-right parser based on the proposed augmented chart (also capable of performing conventional chart parsing) has been implemented and tested in some preliminary experiments. The test data base includes a large number of Chinese word lattices obtained from an acoustic signal processor which recognizes Mandarin speech. Due to the existence of large number of homonyms in Chinese language and uncertainty and errors in speech recognition, very high degree of lexical ambiguity exists in the input lattices. One example of such Chinese word lattice is in Fig. 4. The results show that, all possible constituents for the input word lattice can be constructed and no any constituent needs to be built more than once using the augmented chart parsing. According to the experimental results, the edge reduction ratio (the ratio of the total number of edges built in the augmented chart parsing to the total number of edges built in conventional chart parsing) is on the order of 1/30 ~ 1/80 for our input Chinese word lattices. Although this ratio depends seriously on the degree of ambiguity of the input word lattices, the computation complexity can always be reduced significantly.

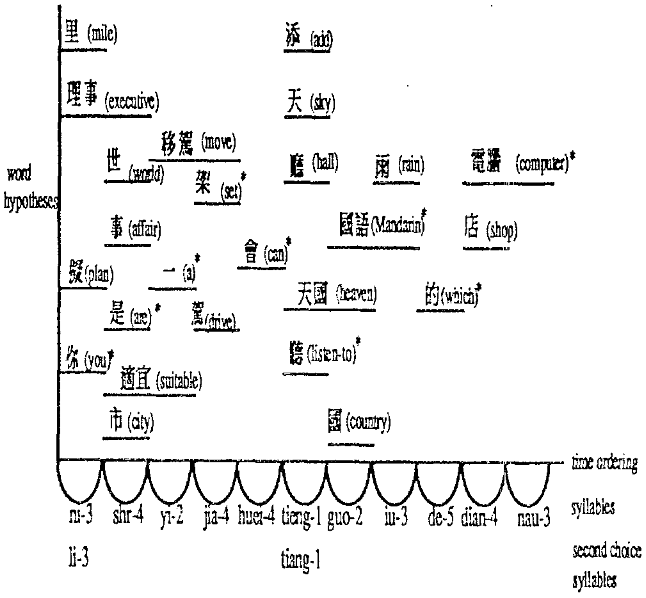

Fig.4 An example in Mandarin Chinese is given here. It is obtained from the Chinese sentence utterance: ni-3 'you' shr-4 'are' yi-2 'a' jia-4 'set' huei-4 'can' tieng-1 'listen to' guo-2 iu-3 'Mandarin' de-5 'which' dian-4 nan-3 'computer' (you are a computer which can listen to Mandarin, 你是一架會聽國語的電腦), where the syllables are represented in Mandarin Phonetic Symbols II (MPS-II) with the integers (1 to 5) indicating the tone. The possible word hypotheses are shown above where the horizontal axis denotes the time ordering of the syllables and the vertical scale shows the corresponding word hypotheses for the syllables, in which only those denoted by "*" are correct words. In this example all the syllables are actually clearly identified and correctly recognized and therefore all word hypotheses are in fact well aligned in boundaries, except that two syllables (the first syllable ni-3 and the sixth syllable tieng-1) are confused by a second candidate (li-3 and tiang-1, respectively). Therefore the ambiguity is primarily due to the large number of homonyms in Chinese language. The line segments under each word hypothesis indicates whether the word hypothesis is composed of one or two syllables. In our analysis, as many as 470 sentence hypotheses are obtained from this example word lattice with most syllables correctly recognized, and the experimental results show that for this example

totally 58132 edges have to be built in conventional chart parsing, while only 925 edges are necessary in the augmented chart parsing. The edge reduction ratio for this example is 1/62.8.

6. Concluding Remarks

In this paper, an augmented chart data structure for word lattice parsing is proposed. It is able to represent and parse a lattice of words very efficiently without changing the fundamental principles, operations and applications of chart parsing. With this proposed approach, all possible constituents of the input word lattice can be constructed and no constituent needs to be built more than once. This will reduce the computation complexity significantly especially when serious lexical ambiguity exists in the input word lattice. It is a general parsing scheme, independent of the grammar formalisms and parsing strategies, thus can be easily extended to different applications. This augmented chart parsing scheme is therefore a very useful and efficient approach for speech recognition applications.

References:

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.

Huang C. R. and Shiu Y. L. (1988) Unification-based Analysis and Parsing Strategy of Mandarin Particle Question. Proceedings of International Computer Symposium, Taipei, pp-38-43.

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

Hellwing P. (1988). Chart Parsing According to the Slot and Filler Principle. Proceedings of the International Conference on Computational Linguistics, pp. 242-244.

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

Lee L. S. et al. (1987). The Preliminary Results of a Mandarin Dictation Machine based upon Chinese Natural Language Analysis. Proceedings of the International Joint Conference on Artificial Intelligence.

Pollard C. and Sag I. A. (1987). Information-Based Syntax and Semantics, Vol. 1. Fundamentals, CSLI Lecture Notes, No. 12., Stanford University.

Stock O.et al. (1988). Island Parsing and Bidirectional Charts. Proceedings of the International Conference on Computational Linguistics, pp. 636-641.

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.

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.