ISCA Archive
http://www.isca-speech.org/archive

First International Conference on
Spoken Language Processing (ICSLP 90)
Kobe, Japan
November 18-22, 1990

A BLACKBOARD ARCHITECTURE FOR A WORD HYPOTHESIZER AND A CHART PARSER INTERACTION IN AN ASR SYSTEM

Andrea Di Carlo (+), Rino Falcone (*)

(+) Fondazione Ugo Bordoni - Via B. Castiglione, 59 - 00142 Rome, Italy
(*) Ist. Applicazione del Calcolo - CNR - Viale del Policlinico, 137 - 00161 Rome, Italy

ABSTRACT

In this work, we present our recent efforts to find an effective framework for the interaction between lexical hypothesization modules and syntactical parsers of an Automatic Continuous Speech Recognition. In this perspective the main problem is the uncertainty processing. We consider the chart a suitable blackboard to represent and to work on the unperfectly recognized linguistic data and structures. Our approach to the uncertainty processing is based on the asynchronous interaction between several modules in which we implement the basic functionalities of the word hypothesization and of the syntactical parsing with different running modalities (top-down and bottom-up knowledge activation, left-to-rigth and bidirectional input processing, breadth-first and depth-first search, best-first and beam use of the scoring). We describe the developed modules and the interaction preliminary framework then we present a brief discussion on the use of the system. Some ideas for the further activity are presented.

I. INTRODUCTION

A very general approach to speech recognition is provided by the assumption that it is possible to recognize by means of the inverse model of the production. For Isolated Words and for the Connected Speech, it is possible to see the recognition process corresponding to a sequential production process. So that it is possible to design a regular recognition model for words which are not necessarily separated by silence. The recognition of a word sequence is possible because a global optimization is applied on the language model [1].

As for as the fluent or spontaneous speech is concerned, the situation is different since it contains noises, bad-structured word sequences, aborted and restarted sentences, etc; moreover, the Italian language is an order free language [2]. In its own nature, the italian continuous speech is not a word sequence: it is very difficult to locate the signal portion promising for the lexical recognition and it is necessary to take every available information and knowledge about speech and natural language into account.

Speech signal is described by means of several parameters which represent the acoustical events characterizing the elementary linguistical units; in this phase, we can detect the transition from physical levels to symbolic ones in the recognition process.

The next step of the symbolic processing of the message is the recognition of words from the system lexicon. The speech nature is such that there is no direct correspondence between the acoustical information in the signal and an unique spoken word. So it is difficult to recognize which word has been produced and emboded into the context of the utterance.

We are finding a solution in the interaction between a word hypothesizer and a syntactical chart-parser in the blackboard framework [3]. The symbolic sequence, given as output by the acoustical-phonetic decoder, is sent to a bottom-up word hypothesization module. It detects the input parts which can be interpreted as lexicon item. Every lexical hypothesis is evaluated and put on to the blackboard, a global data structure with respect the system components. The set of the lexical hypotheses is searched by the parsing subsystem to find a word sequence which is a syntactical correct sentence. The syntactical analysis of a sintagmatic structure can be completed or it can be still active if it represents syntactical structures in which some partial components are missing. Every new structure, both the complete and the active ones, is put on to the blackboard. When the bottom-up parsing is ended, the top-down parsing phase, followed by a word hypothesization phase, is started to complete the active syntactical structure. Then the word hypothesization is constrained over a lexicon subset defined by the syntactical classes and over an input portion. The process continues until a complete syntactical structure over the whole input symbolic sequence is built and entered on to the blackboard or until some stop conditions are reached.

The scheduling is the task of another module. At the moment, it realizes just a sequential running of the system components but, in the next future, it must manage the asynchronous running, as well as the firing events in special situations, it must evaluate, on the basis of some sort of policy, the structures in the blackboard and, finally, it must recognize the stop condition of the process.

II. LEXICAL HYPOTHESIZATION [4]

A word hypothesizer (WH) uses a lexicon implementation and takes in an utterance representation corresponding - in the abstraction level - to the description of the lexical items; it gives out a set of word hypotheses with a similarity degree to some utterance part.

So the hypothesizer generates a word lattice: i.e. an hypotheses' set representing a directed graph. Each word hypothesis is defined by the starting time, the ending time - to locate the hypothesis in the input utterance -, the lexical item indicator - the label - and, finally, the numerical evaluation.

We can distinguish three main steps in the Word hypothesization:

- the segmentation, to locate, in the symbolic input sequence, the boundaries of some portion promissing for the lexical access;
- the access into the lexicon, to retrieve the lexical items leaving from the segment;
- the hypothesis evaluation, to compute the matching degree between the homogeneous representations of the lexical item and the segment.

III. OUR IMPLEMENTATION OF THE WH

We realized two modules for the word hypothesization: an associative hypothesizer and a top-down hypothesizer.

The former segments the input on the basis of an exhaustive selection of the start-markers and of the prediction of the end-markers from the knowledge about the durations of the lexical items; we obtain as many segments as possible from every input symbol and with every lexically possible length.

Then it extracts the hash keys from the segment - all the two symbol substring - and retrieves the words by redundant access into the lexicon. Every hash key is associated to a list of words containing it. So it is possible to increase the activation value of each word in the lexicon when a two symbol substring is found. The lexical hypotheses are created from the localization of the keys in the complete input, from the activated word and from the activation value.

Every activated word is evaluated. We used two possible distance for the evaluation. The first one, the Edit Distance, is a very effective method in framework whose matching problem is seen as a variational problem. The patterns are deformed by local scale transformation: in our case the local transformations are the edit operations i.e. the substitution , the deletion and the insertion of single symbols. The matching is then performed by considering only small pieces of patterns at one time. The pieces are shifted to each other to find the best warping with the minimum number of edit operations. The Edit Distance is very significant but is quite time consuming because many hundreds of matching are necessary and the whole word hypothesization phase becomes too long. To solve the time problem, we choice another distance, less significant but much simpler and faster: it counts the number of corresponding symbols in the matched patterns. The patterns are aligned in a very broad way: by matching the symbols one-to-one without shift until the last symbol of the shorter string is processed.[5,6].

The second one is a constrained top-down word hypothesizer. It directly matches every item of its lexicon against an input segment without bottom-up association. It can accept, from the other modules of processing, the boundaries of the segment on which it must operate. Moreover it uses the subset of the main vocabulary defined by a lexical category as lexicon. Usually, it is used to recover lost hypotheses with less strict automatic constraints and more specific knowledge (shorter input, smaller vocabulary, less strict threshold, specification of the lexical category).

IV. SYNTACTICAL PROCESSING

Given the rule set constituing the language model, a syntactical parser must recognize if a word sequence corresponds to this model (if it is an element of the sentence set generated by this model) and what is its structure. It must completely define the processing and its control.

The classic chart parsing algorithm was designed to parse a sequence of words. In this work, we find an approach to the word lattice processing by development of the ideas of the bidirectional chart parsing [7]. A chart [9] is a directed uncyclic graph representing the state of the parsing. Given a word lattice, the word boundaries of some word hypotheses are called vertices and are represented as nodes in the chart. Each vertex has an arbitrary number of arcs, called edges entering and leaving it. An edge is therefore a link between two vertices and it represents a possible grammatical interpretation of the input portion between the vertices. An edge may be of two types: inactive or active. An inactive edge stands for a recognized constituent. An active edge represents a partially recognized constituent.

The chart is a composition of the symbolic input symbol lattice and an augmented lattice for the structural elements and components that are recognized and built during the analysis process.

Every edge put in the chart is a built structure obtained on the basis of some sort of syntactical rule and of other edges - representing smaller and less abstract structure - yet contained in the chart.

V. THE PARSING SYSTEM

We realized three modules for the chart parsing: a *left-to-right* one, an *island-driven* chart-parsing (both run in bottom-up mode) and a *top-down* chart parser.

The first two modules work on the chart, yet initialized by the lexical items, obtaining the syntactical structures on the basis of a *context-free* grammar. They work both with a *depth-first* strategy but with different analysis direction [8]: the first one is a traditional left-to-right algorithm for which it is necessary to be at the beginning of the structure which has to be analysed; the second one is driven by likelihood island and it is a better solution for the processing of ambiguous, corrupted and fragmented input as the speech. This second module does not only constraint the analysis with the most reliable word hypotheses coming from the lower level of analysis, but it also exclude a combinatorial explosion of attempts.

From the chart built by the bottom-up analysis, the top-down parser find the lexical categories and the temporal boundaries which are suitable to end the incomplete structures corresponding to the more promising active edges.

The syntactical analysis - all the modules - is driven by the precedence degree based on the likelihood of the lexical hypotheses and a likelihood propagation rule for the complex structures.

VI. INTERACTION ARCHITECTURE

It is suitable to underline that the output of the word hypothesizer, the word lattice, and the working data structure of the parser, the chart, are both defined as directed graphs giving the alternate representations - at lexical and at syntactical levels - of input portions.

This analogy is the basis of the interaction between the components: the word lattice can be directly mapped into the chart - the chart has a bigger representation power and it can completely collect the lexical information: it becomes the media for information communications among levels.

In the further generalization, the chart is a data structure global with respect to the system components. From this assumption and from the possibility to realize an opportunistic control,we drew the blackboard inspiration for our system design.

A broad definition of the blackboard model can be the following: there is a global data structure called the blackboard and there are logically independent modules called knowledge sources. The KSs respond to changes on the blackboard: the KSs are self-activating [3]. In our first trial, for more efficiency, we preprogramme the scheduling of KSs as follows.

The symbolic sequence, given as output by the acoustic-phonetic decoder, is sent to a bottom-up word hypothesization module. It detects the input parts which can be interpreted as lexicon item. The word lattice is mapped into the initialized chart: every lexical hypothesis is translated into a chart edge between two nodes corresponding to the word boundaries.

After the chart initialization, the left-to-rigth parser starts; it operates in bottom-up mode, with a best-first strategy of searching in a subspace obtained with all the perfect scored chart edges. If it builds, in a very fast way, a syntactical structure covering all the input and perfectly scored, the success analysis will be obtained and the process will be stopped.

If it obtains no structure covering the input then it will be necessary to continue the process.

The chart is reinitializated at the lexical status and the island-driven parser starts to develop the left-side and rigth-side contexts of every island. The island-driven chart parser builts the correct syntactical constituents concatenating contiguous words or sequence of words to the islands. Some islands cannot be extended because there are no correct words which can be joint to them on the left-side or on the rigth-side. In these cases some active edges are however introduced into the chart.

From these incomplete structures, it is possible to build some prediction on the syntactically possible continuations; when this is possible with only one further step in the syntactical analysis obtaining a structure with only one lexical class missing, a stimulus for a new lexical access is created. The new word hypothesization is constrained over a lexicon subset defined by the syntactical class and over the input portion which as necessary to complete the syntactical structure; moreover this phase uses less strict acceptance threshold for the lexical hypotheses.

With the richer chart, the island driven parsing continues until the end condition is reached. The end condition - fail or success - for this phase is the end condition for all the lexic-syntactical process.

VII. CONCLUSIONS

We can give some indications about the results of our preliminary experiment with the system presented here.

For the word hypothesization module, we obtained a very fast algorithm composed of three procedures: an effective word boundaries' selector, a very fast associative access method and an hypothesis evaluation procedure. This one is quite crucial because it is called many times and it is quite time expansive: that is why we decided to use not the significant Edit Distance and to use the simpler and faster Kohonen distance. In this case there is more confusion in the word lattice with dangerous implication for the global performances of the system.

In our system, we use the typical advantages of the chart parsing: space economy and computation economy. We have on one hand economy of space because the common subparts of built structures are represented only once; on the other hand we have economy of computation because the rules which involve that common subparts need not to be applied as many times as these are involved in the building of bigger structures. For efficiency reason, we have implemented in the chart parsing some dynamical constraints on the edge introduction based on the scoring and coverage rate. The preliminary results show that the blackboard system is able to approach the complex problems of word lattice processing in the perspective of the speech recognition. Specifically, a big problem is that the word lattice, obtained from the word hypothesizer in a continuous speech recognition system, contains some gap patterns: some input portions are not lexically interpreted or there is not the possibility to sequentially connect some lexical hypotheses because their boundaries are too distant.

In general, it is our feeling that our system performances are good when the phonemic errors - performances of the acoustical-phonetic decoder - are distributed quite uniformally in the input and there are not holes in the word lattice. Otherwise we found that our system is able to solve the connection over the holes by predicting some words in them or shifting the boundaries of the contexts when the gaps are isolated in a well interpreted context.

The future efforts are oriented to two main areas: the performance evaluation and the blackboard scheduling definition. The device evaluation will be articulated on the evaluation of the data structure complexity and on the evaluation of the component performances. For the blackboard design, we are developing separate module for the management of the asynchronous scheduling to realize the framework completely.

REFERENCES

[1] L.R.Rabiner, B.H.Juang "An Introduction to Hidden Markov Models", IEEE ASSP Magazine, january 1986

[2] O. Stock, "Parsing Flexible Languages with a View to Recognizing Continuous Speech", Int. Workshop on Advances and Applications of Speech Recognition, Rome, Italy, 27-29 may, 1986

[3] Blackboard Systems, Edited by R. Engelmore & T. Morgan, Addison Wesley

[4] A. Di Carlo, "Word Hypothesization in Continuous Speech Recognition" communication to NATO ASI 90 on Speech Recognition and Understanding Recent Advances, Trends and Applications, Cetraro, Italy, 1-13 july, 1990

[5] A. Di Carlo, A. Paoloni - "An Experiment in Word Hypothesization performed in the Context of a Continuous Speech Recognition System" - Proc. EUROSPEECH 89, Paris, september 1989

[6] Kohonen,Reuhkala - "A very fast associative method for the recognition and correction of misspelt words, based on redundant hash addressing" - Proc. 4th I.J. C. on Pattern Recognition 1978

[7] O.Stock, R.Falcone, P.Insinnamo - " Bidirectional charts: a potential technique for parsing spoken natural language sentences" - Computer Speech & Language, vol 3, pp 219-237, 1989

[8] W.H. Ward & al. - "Parsing Spoken Phrases Despite Missing Words" - Proc. ICASSP 88, april 1988

[9] M.Kay, "Algorithm Schemata in data structure in Syntactic Processing" - Xerox Report CSL-80-12, Palo Alto, 1980