Abstract: Because of the uncertainty involved in terminal recognition, the objective of a syntactic analyzer under uncertainty is no longer to find one structure fitting the speech, but the best one. Two problems must be solved: (1) define and apply an appropriate quality measure of the resulting structures, in order to find the best one, (2) construct and maintain in parallel a set of plausible partial structures, in order to be able to compare them. We present a syntax analyzer which uses the notion of confidence islands realizing parallel construction of the syntactic structures of a speech signal using beam-search strategy based on a specific quality measure we introduced, mixed bottom-up and top-down extension of parsing trees, and unification of islands relating to different parsing trees. Applied to continuous speech understanding, for a 200 sentence single speaker test, the system obtained a correct rate of 90% at sentence level.

1 Introduction

Speech signal is extremely variable and this variability makes uncertain the signal to symbol conversion. A hypothesis at the symbolic level cannot be accepted or rejected at a binary ground. Therefore the recognition of lexical units is unreliable. It is then necessary to take into account the uncertainty in different levels of the interpretation process. In this paper, we propose a syntactic analyzer which plans to compensate for the uncertainty of the signal to symbol conversion.

A common syntactic analyzer leads either to the rejection of barely interfered solutions or to a combinatorial explosion if all possible solutions are built up.

Most of authors propose two main ways to solve this problem: firstly, improving the signal to symbol conversion without any modification of the syntactic analyzer, and secondly, modeling the uncertainty in the grammar by allowing explicitly some errors made during the identification process.

The first method requires taking decisions very early in the interpretation process by using only local knowledge. This includes works in the area of phoneme or word recognition. Due to the intrinsic variability of speech, the recognition rate will be limited. The complexity of the second method increases with the uncertainty of the signal. Furthermore, modeling completely and correctly all possible errors due to the unreliable recognition of lexical units is an enormous work.

These drawbacks lead us to propose another method in which the signal to symbol conversion is only regarded as a coding step in order to reduce the amount of data. Then no deletion, insertion, confusion errors have to be corrected because we only consider the certainty of the different interpretations which are built up in parallel, and we take decisions only at the sentence level.

Two main questions have to be answered: How to define and to apply an appropriate quality measure of an interpretation? How to build and to maintain in parallel a set of plausible partial interpretations in order to determine the best one according to the previously defined criterion?

These aspects will be detailed and a simple example will be given to illustrate our method. We will insist on the following points: How to use the informations collected during syntactic analysis in order to guide the interpretation and how to suppress progressively the uncertainty of the signal on the nature and the position of the terminals.

2 Quality of an interpretation

It is impossible to determine a bijection between a signal portion and a primitive symbol. A same signal portion could often be considered as an image of several symbol classes, but with various membership degrees [2].

The control of the interpretation process and the search strategy require a measure of the quality of an interpretation. The quality of an interpretation measures its degree of coherence with respect to the input speech signal. Since uncertainty originates from and only from signal to symbol conversion, the quality of an interpretation sentence could be defined as the time normalized sum of the likelihood ratios between the symbols of the sequence and the related portion of signal. A likelihood ratio is a similarity measure of the signal to the acoustic image of a primitive symbol.

Let \( \mu(s,t) \) be the plausibility that the signal at time \( t \) is the acoustic image of the symbol \( s \). \( \mu(s,t) \) is also called likelihood ratio of signal to a symbol’s image. The only constraints we impose to this function are:

\[
\mu(s,t) \in [0,1],
\]

\[
\mu(s,t) = 0 \text{ if there is no similarity between the signal and an image of symbol } s,
\]

\[
\mu(s,t) = 1 \text{ if there is a complete coincidence between the signal and an image of symbol } s.
\]

\( \mu(s,t) \) increases with the similarity between the signal at time \( t \) and the acoustic image of a symbol \( s \).

The recognition of a sentence \( i \) means to maximize the sum of individual quality \( u(s) \) of all symbols of the sequence. This quality is defined in terms of the quality of its component symbols \( w(s) \):

\[
Q(i) = \frac{1}{\text{End}(i) - \text{Begin}(i)} \times \sum_{s(i)} u(s)
\]  

(1)

which belongs to \([0,1]\). \( \text{End}(i) - \text{Begin}(i) \) is the temporal length of the considered sentence. We compute \( u(s) \) by:

\[
u(s) = \int_{w(s)} \mu(s,t) \, dt
\]  

(2)

where \( \mu(s,t) \) is the likelihood ratio that \( s \) and the signal at time \( t \) coincide, and \( a(s) \), \( b(s) \) are respectively the starting and ending
3 Principle of the syntactic analyzer

As previously mentioned, a classical syntactic analyzer is unable to analyze structures with uncertain terminals, as in speech recognition or image analysis [1]. Then the objective of a syntactic analyzer under uncertainty is no longer to find one structure fitting the speech, but the best one. An interpretation is the building of one or more trees which describe the syntactic structure of the signal to analyze.

Several authors describe syntactic analyzer capable of starting analysis with any terminal in the grammar [9,10,6,5,7,8]. This technique improves the performances of the analyzer when the detection of terminals is unreliable. We are particularly interested in [8] which develops an algorithm using backtracking and mixing bottom-up and top-down extension of parsing trees. At any time during the analysis process, a tree can be merged with a leaf of another one. The information collected during the syntactic analysis guides the interpretation process and allows to suppress certain ambiguities. The indeterminism is solved by backtracking.

In some situations, it is impossible to make choice from several partial interpretations because of limited knowledge about signal. This ambiguity is due to two reasons: (1) The indeterminism of the language model which indicates several choices at a given step of the interpretation process. For example, if production rules are used, several of them can be activated and it is locally impossible to determine which one could lead to the globally correct interpretation. (2) The uncertainty of the primitive identification. The primitives are recognized using local informations, they could be conflictual with evidences issued from other regions.

Backtracking solves the problem of indeterminism. However, the problem of uncertainty remains because backtracking gives a unique solution to the interpretation. The backtrack-based search follows one solution which locally seems to be reliable, even if a better interpretation could be found by exploiting another alternative. It does not always give the best solution, because of the impossibility of comparing partial solutions at the sentence level.

This backtracking is not sufficient for solving ambiguity caused by uncertainty. Parallel construction of syntactic structures is required if uncertainty is involved in terminals, which means that all plausible choices are developed and that a decision is only taken when it becomes reliable enough. Several authors report that beam search is more efficient than backtrack algorithm.

We implement the structural constructor which uses this strategy as an inference engine with several computing agents [4]. Using the notion of confidence islands, the machine realizes: parallel construction of the syntactic structures of a speech signal using beam search strategy, mixed bottom-up and top-down extension of parsing trees and unification of islands relating to different parsing trees.

4 Method

The system recognizes continuous speech without any pre-segmentation. The basic recognition unit is the phoneme. The function $\mu(s,t)$ previously described is computed. The results of this computational step allow to determine confidence islands to start the interpretation process.

4.1 Confidence islands

A confidence island is a partial interpretation tree with terminals and non terminals whose primitives have been identified with a high quality. It is interesting to initiate reasoning with several confidence islands, because it allows to start with the regions of signal the easiest to identify. Another advantage is that islands give a fragmented but global idea about the structure of the signal, which can be used to guide the interpretation process. The parsing trees of these confidence islands are built up by running backward and forward chaining.

4.2 Mixed chaining

We take syntactic rules as rules in logic inference. The construction of signal structure is finding an inference path with highest reliability. Forward chaining or backward chaining may be used to achieve this objective.

Starting from measurable facts and known facts, forward chaining finds out all facts by applying rules. Since the intermediate problems are fixed arbitrarily, the efficiency of the method is limited by the locality of the information source.

Backward chaining recursively divides a problem into subproblems until measurable facts by successive rule applications. It ignores the possible guidance of the deducted facts to the inference process.

Since it is impossible to verify in due time the large amount of intermediate hypotheses which are eventually confirmed to be invalid later by information coming from some global point of view, separately using both mechanisms will cause low efficiency.

The mixed inference which we use combines together forward chaining and backward chaining. The process starts with several relatively terminals and uses the notion of subproblem to guide the structure construction. Having both intuition and the ability to verify decisions through detection of consistence with known facts, this kind of inference approaches better human intelligence.

In our system, the forward chaining consists in making extension of a syntactic structure from the root of a confidence island. This is of course done for all partial interpretations being constructed in parallel. When several rules can be applied to the root, they are all used and all the possible trees are constructed since they are all locally correct.

The implemented backward chaining adds a tree to a non terminal leaf of an existing syntactic structure. It is performed for all confidence islands of all plausible parallel interpretations. If several rules are applicable for a same node, they are all applied. A node can be extended to the left or to the right in order to add the previous or the following symbol of an interpretation. A backward chaining is performed in two main steps: (1) Given the root of an island and the direction of the extension, this step determines a not yet identified non terminal whose neighbor in the opposite direction of the extension is identified. (2) If the first step succeeds, all the possible rules are recursively applied till a terminal is identified.

The strategy of using forward chaining and backward chaining is as follows: backward chaining is repeatedly performed while it is possible, because this process goes to concrete and incoherence will be discovered as early as possible. In the contrast, the process of forward chaining is stopped once non-terminals suitable for backward chaining are generated, because too much intermediate structures would be generated without the possibility to be confirmed immediately.
4.3 Uncertainty propagation

As all partial interpretations can not be kept (otherwise a combinatorial explosion will happen), it is necessary to define a criterion to drop or to keep a syntactic structure. The maximum number of partial interpretations to be retained at each step of the interpretation process is called the diameter of the beam search. If an island contains only few terminals, there is only few identified information about the neighborhood under uncertainty. It is therefore not obvious that this island could become a correct interpretation. We call such an island fragile. At the opposite, if an island is built with a large number of identified terminals, it is relatively reliable and it is not surprising that it leads to a correct interpretation. The diameter is a function of the number of identified terminals in the parsing tree. Two tables relate the diameter of an island to its fragility and to its minimal quality.

4.4 Tree unification

As previously mentioned, it is necessary to develop in parallel several partial interpretations in order to determine the best one. To avoid maintaining the same substructure in different trees, we need to merge them if these trees are instantiated on the neighboring portions of signal. It is not an easy task, because the algorithm has to detect the substructures in different trees, to verify that these trees can effectively be merged. It must be able to compare correctly the identified region by both trees. The conditions that must be verified for merging two trees are: (1) One structure must be included in the other one or both are producing the same tree. (2) Both trees have to respect the constraint of neighborhood: the identified regions in both trees must have no overlap and no unidentified zone between both regions.

Examples of situations requiring a merging are given in Figure 1. The first example illustrates the constraint of neighborhood and equality and the second one gives two cases of inclusion.

5 An example

We give an example which illustrates how the syntactic analyzer proceeds. In Figure 2 we give the grammar we use in this example. The test sentence is "We appreciate much the brilliant writer we met yesterday".

Initially, the lexical level proposes 4 words. They constitute the 4 confidence islands allowing to start the analysis (Figure 3). A forward chaining produces new islands. No node has a brother (Figure 4), it is impossible at the moment to use the backward chaining. Another forward chaining is therefore performed (Figure 5), so that a backward chaining can be made (Figure 6). Figure 7 shows a situation requiring the application of the merging algorithm, in Figure 8 result of the merging is given. Finally, Figure 9 gives the correct interpretation.

6 Experiment

The proposed method was applied to continuous speech understanding using the phoneme as basic recognition unit. The language used is Chinese. The number of phonemic units used is 50. The speech signal is at 16kHz sampled and with a 12 coefficients LPC analyzed. The task consists of office robot control and railway information request application defined by a context-free grammar with about 250 rules and 200 terminal words. For a 200 sentence single male speaker test database, the system obtained a correct rate of 90% at sentence level with a reasonable computational load on a SUN workstation [2]. Experimentally, the complexity of this constructor is about $O(n^3)$, similar to that of equivalent syntactic recognizers. Because ambiguous grammars are accepted, this complexity is not excessive.

7 Conclusion

It is necessary to use constraints in order to reduce uncertainty of the signal to symbol conversion. The syntax is one of these constraints and we propose syntactic structures interpretation under uncertainty.

We show that uncertainty and indeterminism are of different nature and must be solved with different methods. We indicate that the terminal identification have to be realized during the syntactic analysis and that uncertainty must be removed progressively. Therefore we propose a set of algorithms which perform parallel syntactic structures construction with a mixed chaining algorithm. The interpretation process is highly driven by the information collected during the analysis, contrary to traditional continuous speech recognizers, no pre-segmentation is performed. We defined a quality measure directly based on the quality of low level signal to symbol conversion. Experiments show that the method is robust.

We conclude that our approach contributes to the solution of efficiently using the information obtained during the parsing phase to guide the speech understanding process and of progressively removing the uncertainty about terminals.

References

Figure 1: two examples of possible situations requiring the application of the merging algorithm. (indicated by "v")

Figure 2: Grammar rules used in the example

Figure 3: initial lexical hypotheses consisting of 4 words, each forms one confidence island at the syntactic level

Figure 4: the forward chaining generates islands in each of which no brother exists, a backward chaining is thus impossible.

Figure 5: another forward chaining gives trees with brothers (108, 109 and 110). A backward chaining is to be applied.

Figure 6: A backward chaining to the left creates, for islands 108, 109 and 110, respectively 1, 1 and 3 partial interpretations, due to the indeterminism of the grammar rules. Only one tree is confirmed by low level evidence.

Figure 7: configurations requiring an unification

Figure 8: result of the unification

Figure 9: All portions of speech signal examined and root being the start symbol of the grammar, the recognition process is terminated.