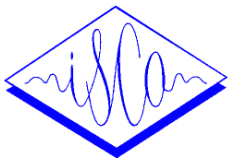

2nd European Conference on
Speech Communication and Technology
EUROSPEECH '91
Genova, Italy, September 24-26, 1991

Improving Speech Understanding Performance through Feedback Verification*

P. Baggia, L. Fissore, E. Gerbino, E. Giachin, and C. Rullent

CSELT - Centro Studi e Laboratori Telecomunicazioni
Via Reiss Romoli, 274 - 10148 Torino, Italy

Abstract

A parser for continuous speech has to deal with lattices where the word hypotheses of the correct sentence are not usually perfectly aligned and short function words can be missing. To cope with these problems, a two-way interaction between the recognition module and the parser, called *feedback verification procedure (FVP)*, has been investigated.

The parser generates many solutions, that are fed back to the front-end processor (FEP) which realigns them against the acoustical data, finds the missing function words among the given candidates and attributes them a new score. The best scored solution is finally selected by the parser.

1 Introduction

A successful approach to spoken language understanding involves a two-level architecture in which acoustic-phonetic processing is separated from linguistic processing, the interface between the two levels consisting in a so-called *lattice of word hypotheses*. In this architecture, the flow of information is unidirectional, from the FEP to the parser. One of the advantages featured by this setup is that it permits to design fast parsing algorithms that are able to process a lattice (i.e. to find the best-scored sequence of word hypotheses consistent with the language model) without requiring the presence of short function words, which are often unreliably recognized or even missing from the lattice itself.

Parsing lattices involves a drawback, however. It is extremely unlikely that word hypotheses making up the correct sentence are perfectly aligned: gaps and overlaps will be observed between them virtually all the times. This means that the acoustic information of small portions of the waveform is not exploited, and may lead the parser to find a wrong solution. Also, function words may be sometimes essential to correctly understand the meaning of a sentence. In order to cope with these problems, a two-way interaction between the FEP and the parser has been investigated, called *feedback verification procedure (FVP)*. In this configuration, the parser is run until it finds many different sentences, possibly containing multiple possibilities in place of missing function words. These sentences are then fed back to the FEP, which realigns them against the acoustic data and attributes them a new score. The best-scored solution is then selected as the "correct" one. In addition, the FEP also finds the best-matching candidate for function words that were missing in the lattice. Experiments show that this method increases the rate of correct sentence understanding, as well as the number of function words correctly recognized.

*This research has been partially supported by EEC ESPRIT project no. 2218 SUNDIAL.

2 The Verification procedure

The FVP has two main goals: the first one is the ability to detect short function words with a good accuracy even when they are not present in the lattice; the second one is an increased average understanding rate. The advantage of FVP is a still limited amount of interaction between recognition and understanding, thus avoiding inefficiencies that can be critical for real time systems.

There are two parsing modalities: in the first one the parser stops at the first solution while in the second one it continues until the predefined resources are consumed. The Verification procedure requires the parser to work according to the latter modality. The analysis continues until one of the following conditions are satisfied: no more heap memory to allocate phrase hypotheses is available, a predefined time interval has been expired or a predefined maximum number of possible solutions have been generated.

The FVP analyses all the solutions generated by the parser, one after another. The first step consists in the generation of a word graph for each solution, taking into account the solution parsing structure generated by the parser; the second step involves the acoustic verification function, which orders the sentence hypotheses, provided by each word graph, according to their global acoustic scores. The third final step involves the selection of the best word sequence and the completion of the parsing activities, like the generation of the internal semantic structure. Let us first examine the FVP with a greater detail and then discuss the kind of problems it can solve.

2.1 Word graph generation

During the lattice analysis the parser is able to skip, when necessary, short function words with low semantic contents by using dummy temporal placeholders [1]. So a solution

does not include all the uttered words but contains all the relevant morphological, syntactic and semantic information that are necessary to determine the possible candidates. For instance, given the utterance "Leggi il messaggio di Rossi" ("Read the message of Rossi"), the parser will probably find a solution like "Leggi ?? messaggio ?? Rossi" where the ?? are time placeholders that stand for a word missing from the lattice. From the syntactic rule associated to the constituent "?? messaggio" it is possible to obtain the admissible word candidates. Such dependency rule will be like: *NOUN = ART **, i.e. where an article depends on the noun; in fact a rule like: *NOUN = PREP ** has been activated to generate a constituent but such interpretation of the constituent is not part of the solution.

The candidates are articles and the morphological agreement conditions on gender and number that have been added to the dependency rule require it to be singular and masculine: the possible candidates are "il" (the), "lo" (the), "un" (a). Note that the second candidate is not congruent with a word that begins with an "m", in fact the two articles "il" and "lo" are both masculine and singular and which is used depend on the first letter of the noun. We implemented a few of such phonological rules. The word graphs generated by the parser are then quite simple. There is a node for the beginning of the sentence, a node for each inter-word position and finally a node for the end of the sentence. Arcs are only between adjacent nodes and correspond to word candidates.

2.2 The Verification of word graphs

Each word graph, corresponding to a solution, is analyzed by the FEP independently from the other. The result is a sequence of words for each word graph; a word is chosen among word competitors each time in the word graph there is more than one arc connecting two adjacent nodes; of course the selection is made with the aim of having the highest possible global score for the sequence. An important side effect is just this global score that is assigned to the word sequence.

As far as the real implementation is concerned, the sentence hypotheses are found out by each word graph and are automatically compiled into a string of context-sensitive phonetic units, modelled by HMMs, and organized in a tree structure. This allows to speed up the decoding process, as the initial common subsequences of the sentence descriptions are shared. The Viterbi acoustic-decoding, relying upon a Beam-Search strategy, computes the likelihood score for each sentence, by observing the frames of the input speech [2].

3 Reasons for the verification procedure

There are two main reasons for the FVP; one is connected to the reliability of the score combination strategy used by the parser and the other is related to the fact that very short words are often missing from the lattice and, if present, have a very low score reliability.

3.1 Reliability of score combining methods

As the lattice contains a large number of word hypotheses the parser could possibly extract from it many different word sequences that satisfy all the available constraints (time alignment, morphology, syntax, semantics and pragmatics).

An exhaustive search in the lattice is not performed, given the large amount of resources that would be required. The parser is guided by scores and uses a search strategy, described in [3], that is almost optimal, i.e. that when a solution is found there is a good probability that no better solutions can be found by continuing the analysis.

Of course the [almost] optimality holds given the score combining method used to assign a score to a phrase hypothesis given the scores of the supporting word hypotheses; we use the density method [4].

A problem is related to the difficulty of having, for each word in the lattice, a score sufficiently reliable when used to compare word hypotheses of different length and covering different speech portions. Sometimes long words (or short ones) are penalized over the others; in addition hypotheses pertaining to a certain speech portion can be penalized over those covering different speech portions. The feeling is that only when the same speech portion is covered, scores are really comparable. Through FVP the parser first generates the best N candidate solutions using a standard score combining method, then the final solution is selected using the new scores assigned by the FEP. As all solutions refer to the same time interval, now scores are comparable in a more reliable way.

3.2 Short words missing from the lattice

For certain applications short function words may convey a meaning not essential to generate a formal representation of the utterance meaning suitable to perform the correct action. In other cases this is not the case: for instance in the sentence "i treni da Torino a Roma" (the trains from Turin to Rome) the two prepositions are quite short but nevertheless quite relevant to determine if Roma (Torino) is the departure place or the arrival place. The FVP can detect the correct words even when they are not in the lattice.

Short words are often missing from the lattice (or at least they are not really reliable) because the attempt to generate them means a very large number of word hypotheses in the lattice and the shortest of them will always be missing. In addition there is the problem of assigning to them reliable begin and end points.

The FVP creates the ideal conditions to find these words with sufficient reliability: for each time placeholder a very limited number of candidates are proposed (usually less that five) and the previous word and the following word are often normal reliable words. At this point the FEP can detect the word with quite a greater reliability and co-articulation rules could also be used. Note that we do not require the short word to start at the frame where the previous word ends: during the FVP all the words can found their new best time alignment as to maximize the global score. After the FVP has detected the correct short word it will be possible to obtain the desired formal structure for the utterance meaning.

3.3 Time alignment and short words

Another problem is the reliability of the process of combining together the scores of words when there is not a perfect

alignment between adjacent words. Let us suppose that the sentence "... Roma a Perugia" (... Rome to Perugia) has been uttered; the word "a" (to) is a short word, missing from the lattice (see Fig. 1), and in the lattice "Perugia" is partially overlapped by an extraneous word hypothesis "Foggia" (all Italian city names); in addition "Foggia" has a better score than "Perugia" and the temporal gap between itself and "Roma" is within the threshold used for missing short words.

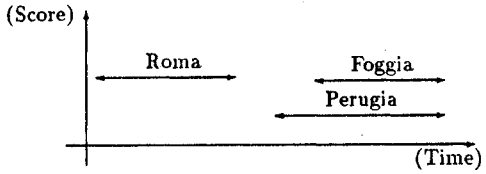

Figure 1: A lattice example.

The first solution generated by the parser will be something like " ... Roma ?? Foggia". Let us suppose that linguistic knowledge gives "a" and "per" (to) as plausible candidates to fill such gap. Now we can see how the FVP can reject "Foggia" and resume "Perugia". Supposing "Roma" perfectly spotted, there are four possible couples of words that can follow: (a Foggia), (per Foggia), (a Perugia) and (per Perugia).

While the score of "Foggia" is better that "Perugia", they do not start at the same frame: "Foggia" is shorter than "Perugia". On the contrary, the various couples do need to cover the same speech portion (that starts at the ending frame of "Roma"). The speech portion of the uttered "a" can fill exactly the gap between "Roma" and "Perugia" while for the couple (a Foggia) the word hypothesis "Foggia" needs to be "stretched" to cover, together with "a", the larger gap: that could lead to a global score worse than that of (a Perugia). As the word "per" has not been uttered we can assume that both (per Foggia) and (per Perugia) do not compete for the best score.

4 A drawback and a possible correction

The obvious disadvantage of the method is that the parser must continue the analysis even when a first solution has been found, then reducing the global system efficiency. Efficiency and accuracy usually require contrasting efforts but there are ways to deal with this problem. The first way is the obvious one: to have an intrinsically efficient knowledge representation [5] that integrates syntactic and semantic knowledge [6] and an efficient parsing strategy [3], suitable to be parallelized, if necessary [7]. This led to a parser that is able to generate a few solutions within a reasonable time interval.

A second way is the idea of using the best word sequence given by the FEP in addition to the lattice. Sometimes the best sequence is correct, sometimes it is not. The parser first tries to analyze the best sequence by seeing it as a quite simple lattice; if a solution is found it is assumed to be correct and the system accepts it. Otherwise the analysis continues on the real lattice for the fixed amount of resources and a few other solutions can be obtained; at the end the FVP takes place. The advantage is from one side that for a certain percentage of sentences (about 42% with discrete HMM) it is not necessary to perform the analysis on the lattice, with the consequent gain in the average efficiency. From the other side there is also a small increased accuracy (about 1.7%).

5 Experimental results

Experiments have been performed starting from a test sentence set of 600 different sentences. Ten speakers have uttered 60 sentences each, producing speech material on which all the experiments are performed. We present here the results derived from the use of two different acoustic-phonetic decoding techniques:

- a discrete density HMM (DDHMM) Viterbi algorithm based on the use of 305 context-sensitive acoustic-phonetic units (R0),
- a continuous density HMM (CDHMM) Forward decoding using a mixture of 45 gaussians based on the use of the same phonetic units (R1).

Besides 27 context-independent phonemes, the set of units includes 58 phones in function-words (articles, prepositions), 107 triphones and 113 right-context biphones; triphones and biphones were selected according to their frequency of occurrence in a training corpus made up of 8800 sentences [8].

First order, left to right Hidden Markov Models with 3 states without any skip transition are used to characterize the units both for DDHMMs and for CDHMMs. The DDHMM Viterbi decoding algorithm is working real-time on the real machine while the CDHMM Forward one is only the result of a simulation phase and it takes a long time. The FEP generates both a lattice of word hypotheses and the best sequence. The best sequence is in the form of a set of word hypotheses too, now covering exactly the whole utterance speech signal without gaps or overlaps. Three different parsing modalities have been experimented: using only the lattice (C0), trying first the best sequence and, only in the case of failure, the lattice (C1) and finally inserting the best sequence, as a set of word hypotheses, into the lattice (C2).

For all the three cases the percentage of correct understanding has been computed both before and after the FVP. This measure is similar to the sentence recognition rate but it differs when short function words are involved: the sentence under analysis is considered correct if it differs from the test sentence only for those short function words that the parser is able to skip during the analysis. The results are reported in Tab. 1.

These experiments show that, thank to FVP, it is possible to gain a relevant reduction of the error rate for R0 (19.1%) while in the case of R1 this reduction is more limited (6.1%) because R1 is more precise to detect the correct beginning and ending points for the word hypotheses.

The experiment R0-C2 shows the ability of FVP to find short words in a reliable way (see Tab. 2). The 600 test lattices have been classified in the following way before and after the FVP. Let us use the term PC (perfectly correct) in the case where all the sentence words have been identi-

fied correctly; the term C (correct) is used when all but the function words have been identified correctly and a placeholder has been inserted exactly where the short function word has to be; the term AC (almost correct) refers to the previous case but when there is not such exact correspondence. The others are referred to as IC (incorrect or failed).

Out of the 299 sentences that had the chance of being corrected by FVP 259 (86.6%) were able to fill all their

Experiment	FVP	Corr. (%)	Fail. (%)	Incorr. (%)	Error reduction
R0-C0	NO	65.8	4.8	29.8	-
R0-C0	YES	70.3	5.7	24.0	14.4
R0-C1	NO	68.6	4.2	27.2	-
R0-C1	YES	72.0	5.2	22.8	10.8
R0-C2	NO	67.0	10.5	22.5	-
R0-C2	YES	73.3	11.3	15.5	19.1
R1-C0	NO	79.2	9.8	11.0	-
R1-C0	YES	79.9	9.8	10.3	3.4
R1-C1	NO	83.5	6.2	10.3	-
R1-C1	YES	84.5	6.2	9.3	6.1
R1-C2	NO	81.5	9.5	9.0	-
R1-C2	YES	82.0	9.5	8.5	2.7

Table 1: Experimental results for two recognition algorithms (R0, R1) and three parsing modes (C0, C1, C2).

FVP	PC		C		AC		IC	
		%		%		%		%
bef.	56	9.3	299	49.8	47	7.8	198	33.0
aft.	381	63.5	0	-	58	9.7	161	26.8

Table 2: The ability of FVP to find short function words

placeholders with the correct function words, thus leading to a PC sentence. The average number of placeholders for each sentence is 1.61. The other 40 (13.4%) sentences came to the following conclusions: 23 (7.7%) sentences found at least one incorrect function word; 10 (3.4%) sentences either failed or became incorrect after verification (the order is changed); for the other 7 (2.3%) sentences the recognition algorithms preferred to select the no-candidate option given by the parser when appropriate.

6 Conclusions

The verification procedure means a tighter interaction between recognition and understanding modules, but the load of interaction is not excessive as inefficiencies that arise when a lot of work is performed on constituents that are not likely to lead to a solution are avoided. In a first phase of parsing where somewhat relaxed constraints are used to direct the parsing towards the most promising directions, generating a set of candidate solutions with a good probability of including the correct one. At this point more precise constraints (like precise time alignment and roles of function words) are used to perform the second phase: the selection of the best candidate in the set.

The experimental results were encouraging: it has been possible to gain a relevant reduction of the global error rate for the discrete recognition algorithm (19.1%) and a still acceptable one in the case of the continuous one (6.1%). It was also possible to detect the words missing from the lattice with a good accuracy as only 7.7% of the word graphs resulted in an incorrect word sequence by picking up at least one incorrect short functional word.

Acknowledgements

We acknowledge Roberto Pacifici and Davide Clementino who, in the real time system, respectively implemented the compilation of the string of context-sensitive phonetic units and the Viterbi acoustic decoding.

References

[1] E.P.Giachin and C.Rullent, "Robust Parsing of Severely Corrupted Spoken Utterances", *Proc. COLING 88*, Budapest, August 1988.

[2] A.Ciaramella, D.Clementino and R.Pacifici, "A PC housed Speaker Independent Large Vocabulary Continuous Telephonic Speech Recognizer", *Proc. EUROSPEECH 91*, Genova, September 1991.

[3] E.P.Giachin and C.Rullent, "Linguistic Processing in a Speech Understanding System", *Proc. NATO Workshop on Speech Recognition and Understanding*, Cetraro (Italy), July 1990.

[4] W.A.Woods, "Optimal search strategies for speech understanding control", *Artificial Intelligence*, vol. 18, 1982.

[5] P.Baggia, E.Gerbino, E.P.Giachin and C.Rullent, "Efficient Representation of Linguistic Knowledge for Continuous Speech Understanding", *Proc. IJCAI 91*, Sidney, August 1991.

[6] M.Poesio and C.Rullent, "Modified Caseframe Parsing for Speech Understanding Systems", *Proc. IJCAI 87*, Milano, August 1987.

[7] E.P.Giachin and C.Rullent, "A Parallel Parser for Spoken Natural Language", *Proc. IJCAI 89*, Detroit, August 1989.

[8] L.Fissore, P.Laface and G.Micca, "Comparison of Discrete and Continuous HMMs in a CSR Task over the Telephone", *Proc. ICASSP 91*, Toronto, May 1991.