gefördert vom
Bundesministerium
für Forschung
und Technologie

Verbmobil
Verbundvorhaben

An Investigation of Tightly Coupled Time Synchronous Speech Language Interfaces Using a Unification Grammar

Hans Weber & Andreas Hauenstein

IMMD VIII/UER & NATS/UHH

Vm
Report 9
March 1994

March 1994

Hans Weber & Andreas Hauenstein

IMMD VIII – Künstliche Intelligenz
Universität Erlangen-Nürnberg
Am Weichselgarten 9
D-91058 Erlangen
&
FB Informatik, AB NatS
Universität Hamburg
Vogt-Kölln-Str. 30
D-22527 Hamburg

Tel.: (09131) 699 117
(040) 54715 522 -
e-mail: `@weber@faui80.informatik.uni-erlangen.de`
`andreas@nats1.informatik.uni-hamburg.de`

Gehört zum Antragsabschnitt: 15.3: Phonologisch informierte akustische Analyse, 15.7: Architektur integrierter Parser für gesprochene Sprache

Das diesem Bericht zugrundeliegende Forschungsvorhaben wurde mit Mitteln des Bundesministers für Forschung und Technologie unter dem Förderkennzeichen 01 IV 101 H9 gefördert. Die Verantwortung für den Inhalt dieser Arbeit liegt bei dem Autor.

Accepted under the same title at the AAAI Workshop on Integration of Natural Language and Speech Processing held at the Twelfth National Conference on Artificial Intelligence (AAAI-94), Seattle, Washington, USA on July 31st/August 1st, 1994.
To appear with slight modifications at the CONVENTS 94 in Vienna.

Abstract

This paper reports on some experiments on time synchronous interfaces between word recognition and parsing, performed with a beam decoder and a chart parser. Using the same acoustic models, language model, and unification grammar, bottom-up and two interactive protocols were implemented and examined. Results show that close integration is possible without unbearable time penalties, if restrictions from both modules are applied to focus the search process.

1 Introduction

Integration of speech and language technology has been of growing interest for a couple of years. A variety of interfaces has been introduced between acoustic and linguistic processing. In this article, we concentrate on some as we think prototypical variations of the time synchronous strategies. There are several reasons why these strategies are of special interest:

- Humans seem to do acoustic–linguistic processing in a tightly coupled manner.
- A partial syntactic analysis of the word sequence exists while it is being uttered, permitting interruptions and early detection of misunderstandings, and allowing higher modules to begin their analysis early on.
- Restrictions from syntactic knowledge can be used to influence the acoustic search process to increase recognition accuracy.

We leave the knowledge bases, i.e. HMMs, n-gram model and unification grammar constant, in order to achieve true comparability. Being interested in the corresponding search problem, we couple a beam decoder and a chart parser in three different ways, the first of which is strictly bottom-up, the latter two being interactive architectures.

Murveit et. al. 90 [7] and Dupont 93 [2] presented promising results on a time synchronous coupling of the decoding step with finite state networks and context free grammars. The common approach in natural language understanding is unification grammar parsing, which presents a much harder parsing problem but allows for a more thorough semantic analysis. In search of a general solution to the integration problem of speech recognition and natural language understanding we are looking for a well working combination of the standard technologies.

In the first experiment — the bottom-up arrangement — the decoder produces a large connected word graph using only the acoustic knowledge source. The lattice is incrementally sent to the parser as it ist produced hypo by hypo. The parser searches from left to right using the language model and unification rules simultaneously. In all experiments decoder and parser run in parallel.

In the two interactive architecture experiments the modules exchange bottom-up as well as top-down information about word hypotheses. In the first setting, the decoder produces hypotheses based on their acoustic score. The parser receives the "lookahead" of the decoder and provides language model information and a *yes or no* based on the unification grammar in a verification step, which is passed back to the decoder and used for rescoring.

The final experiment is a configuration where the parser produces predictions from left to right, based on the unification grammar and the language model. These predictions are used as restrictions before the acoustic match is done by the decoder, the latter always being one step behind the parser.

The results are formulated in terms of word recognition rate and recognition rate for whole utterances, using the highest scoring string identified by all knowledge sources in combination. Besides this we report on efficiency matters, as there are cost of communication, synchronization of the modules and native processing requirements of the decoding and parsing steps in the different experiments.

2 Components Used in the Experiments

This section gives a short description of the acoustic decoder and the parser used in the experiments. The communication between the two components was implemented with a socket based network communication package we developed for that purpose. It allows processes parallely running on different machines to exchange string messages.

2.1 The Decoder

We use an HMM based word recognizer which relies on a pronunciation dictionary representing each word as a unique sequence of phoneme–like subword units. The phoneme models are context independent. Their design is similar to those used in Lee 89 [6], i.e. 7 states and 12 transitions with discrete emission probabilities associated with the edges.

The feature set consists of 5 PLP coefficients, as described in Hermansky 90 ([4]), log energy, 5 delta PLP-coefficients and delta log energy.[1]

The search procedure is a time synchronous beam search, similar to the approach described in Ney 92 [9].

We adapted the search procedure to output word hypotheses on the fly. Word transition scores received with top down messages can directly be incorporated into the viterbi search.

Received word transition scores are kept and propagated to future grid points to avoid redundant communication.

2.2 The Parser

The parser is an active chart parser, which essentially performs a viterbi search based on n-gram probabilities. We started from a basic parser, similar to the one described in Chien et al. 93 [1], extending it with respect to strict left to right processing, pruning, preprocessing of static rule series, efficient processing of unification grammar rules and fast prediction of subsequent word hypotheses.

2.2.1 LR-Parsing with the Active Chart Parser

The initial chart consists only of the vertex zero, with the starting edge and subsequent left corner edges. Whenever a word hypo is received, the chart is filled with vertices - one per acoustic frame - up to the hypo end. In one cycle all hypos are read which end at the next frame. Then pruning is performed on the agenda. The configurations which survive pruning are parsed until the agenda is empty.

[1]Thanks to Kai Hübener (University of Hamburg, Comp. Sc. Nat. Lang. Div.) who gave us both helpful comments and HMM and PLP implementations.

2.2.2 Pruning

Each word hypothesis is associated with an acoustic log probability. The initial edge has a zero score (probability 1). Whenever we combine two partial paths to a new one, we add the n-gram transition log probability to the scores of the original edges to get the new edge score. Since we introduce new empty edges in new vertices only, if a partial path exists up to this vertex which leads to a top down introduction of the new edge, such an edge receives the total log probability of the partial path. When the parser is running left to right in cycles as described above, the scores of all pairs of edges on the agenda result from the same portion of the signal, namely from the start to the actual vertex (frame). In this way we can directly compare them and prune with an offset from the maximum in each cycle. The parser can also be run in non-incremental mode. Then we normalize by the number of additions which took place, which correspond to the number of frames and words, to be able to judge partial paths of different length.

2.2.3 Rule Application

Parsing top down, we use an efficient modification of the Restriction Mechanism proposed in Shieber 85 [10]. Restrictive linguistic information such as part-of-speech and subcat information are encoded as global types of feature structures, which can be compared and unified in O(1). A context free backbone is used in top down rule insertion, where the types are used as grammar variables.

2.2.4 The Unification Grammar

The unification grammar we used in all the experiments consists of a lexicon whith 363 word form entries showing an average ambiguity of 1.3 and a set of 47 grammar rules. A grammar rule is a feature structure which describes a context free rule where variables are replaced by typed feature structures and additional restrictions which must hold between them. A simplified example of a rule is given below. The rule describes word combinations of German where the *Vorfeld* of a sentence with an infinitival complement is filled with an argument.[2].

[2]Thanks to Lutz Euler for developing the formalism

```
rule[(name [])
     (rule <
        sentence[(syn %3)]
        %1=vorfeld[(syn [(wh no)])]
        vmodfin[(syn %3=[(status 1)
                 (subcat <%1 . %2>)])]
        vp[(syn[(status 1)
                (subcat %2=[])] )]
        >)]
```

The grammar covers the 200 sentences of our train information corpus.

2.2.5 Prediction

When run with prediction of new word hypotheses, as in the third experiment, for the current ending vertex, the bigram produces the best extensions of the paths ending in that vertex. The set of candidates is pruned before its members are tested for their ability to form an analysis with one of the active edges ending at the current vertex.

2.2.6 Verification

In the second experiment, every word hypothesis is verified when it is built into a search path for the first time. The verified hypo is guaranteed to be the best one which satisfies the unification grammar due to the best first search performed. The verified hypo is given back to the decoder, annotated with the n-gram value.

3 The Speech–Language Interface

Word hypotheses are passed from the recognizer to the parser as a word graph. Each edge carries a word label, the start and end time of the corresponding acoustic hypothesis, and the score per frame of the acoustic observation. Each word on an edge entering a node may be followed by any of the words on the edges leaving that node.

The graph is represented as a sequence of messages of the form

```
<startnode> <endnode> <word> <score>
<starttime> <endtime> <flag>
```

The flag field is used to supply information about the status of the message, e.g. whether the language log probability of a hypo is already known by the decoder in verify mode.

During recognition, these messages are sent from the recognizer to the parser time synchronously (and the other way round).

The parsing process is guided by the bigram probabilities as well as by the acoustic scores. Some care has to be taken with time normalization to make sure it is possible to compare and combine the scores of edges of different lengths. The score sc of an edge spanning a sequence of words is given as

$$\mathrm{sc} = \frac{c_A \sum s(w) l(w) + \sum log P(w|v(w))}{\sum l(w) + len(E)} \tag{1}$$

where all the sums span the words w in edge E. $v(w)$ is the word preceding w, $s(w)$ is the acoustic score per frame of w, $l(w)$ is the number of frames spanned by w, $P(w|v(w)$ is a bigram probability, and $len(E)$ denotes the number of words spanned by edge E. c_A is an empirically determined weight to balance the relative influences of the bigram score and the acoustic match. c_A was set to the same value in all experiments.

4 Interaction Strategies

We examine three strategies of time synchronous interaction between the acoustic decoder and the parser, namely incremental bottom up (BU), prediction mode (PR), and verification mode (VR).

4.1 Time Synchronous Bottom Up Interface

This is the most straightforward way of coupling the word recognition module and the parser. Each word end hypothesis encountered by the decoder is sent to the parser in the format described in section 3. There is no feedback whatsoever from the parser to the recognizer. Still, this mode of operation is intrinsically different from a sequential approach where linguistic analysis starts only when acoustic analysis is finished. While the best acoustic match in BU mode is exactly the same as with a standalone decoder, the string accepted by the parser may be totally different. In fact, the word sequence accepted by the parser is the best sequence according to the metric described in section 3 that will parse, no matter how far down the acoustic n–best list it is.

The lack of top–down information has advantages as well as drawbacks. On the one hand, no complicated synchronisation mechanism is necessary. Thus the parser is free to prune and throw away as many hypotheses as it likes without affecting the acoustic decoding process. One way to make use of this advantage is to use a coarser timescale by choosing only one among all the acoustic realizations of each word ending inside a window of n frames. This is not possible for the PR and VE interfaces, and presents a real challenge there. On the other hand, the decoder is of course unguided by what is parsable and by what is not. This means that the pruning threshold is always determined by the best *acoustic* match, which may cause undesirable pruning of acoustically less plausible, but parsable hypotheses. In some of our experiments the effect of this was that the parser was constantly swamped with words that could not be integrated into any of the current parses and spent most of its time verifying their unparsability over and over again.

4.2 Top Down Predictions from the Parser

LR Prediction (PR) by the parser is an approach which has been tried before, using a Tomita-Parser (e.g. Kita et al. 89 [5]). We were especially interested to find a way to compute predictions of word hypotheses using an active chart parser (ACP) and a unification grammar – on the one hand to show that this is possible at all, on the other hand to achieve true comparability with our BU and VR strategies. The computation of predictions with our ACP exploits the framewise pruning and parsing scheme described above. After every cycle on a vertex/frame i, we find an active edge j for every word hypo k ending in vertex i, iff that hypo k both survived the beam and was built into a partial parse. The future successors of hypo k are supposed to be able to combine with edge j or an empty active edge of vertex i, which was introduced by edge j in a seek-down (predictor) step. We propose a generate-and-test algorithm:

1. Let P be $\emptyset$
2. With hypos k, for which an edge j exists, let S be the set of the n best n-gram successors.

3. Create a vertex -1 and -2

4. For all edges$j$ and for all empty edges in vertex i create an active edge in vertex -1.

5. For all s in S add an inactive edge spanning -1,-2, and parse it one step. If one active can apply successfully, add s to P.

A prediction of the same wordform together with the best n-gram score from a predecessor is sent to the decoder only once for a vertex i. This permits the decoder to perform a standard viterbi search with the n-gram scores included. The important difference to isolated decoding with acoustic restrictions and n-gram only is that for a given frame the decoder will start only those models, which are predicted by the parser and could — according to the knowledge of the prefixes — lead to a complete parse.

The effects are twofold. The search space of the decoder is reduced and as a result, the search space of the parser to the right of a prediction is reduced, too. Due to the focussing of the decoder on parsable continuations of search paths the amount of word hypos to be handled by the parser is much smaller.

4.3 Verification of Word Hypotheses by the Parser

In PR mode, it is the parser who looks ahead and supplies the acoustic search with the next words to hypothesize. In VR mode, the decoder is one frame ahead and passes acoustically plausible hypotheses to the parser for syntactic verification. Thus, the time–consuming task of generating predictions is no longer necessary. The drawback is that the decoder cannot take advantage of the bigram probabilities immediately, but only when the end of a word has been reached. This also affects the implementation quite strongly and makes the interface different from the usual way of integrating language model probabilities. A similar problem has to be solved by systems using a tree–structured pronunciation dictionary [8].

The communication loop in the decoder looks as follows:

```
-------------------------------------------
FOR each word w ending at time t
  IF w with this start time ts has not been
      verified earlier THEN
        send w to the parser;
```

```
  ELSE
      w.score *= w.pscore[ts];
  ENDIF
ENDFOR

WHILE parser sends triples (w,ts,pscore)
  w.score *=  pscore;
  w.pscore[ts] = pscore;
  remember that (w,ts) has been verified;
ENDWHILE

FOR all word ends w which were not verified
  w.score := 0;
ENDFOR

send the remaining verified unsent
words w to the parser
-------------------------------------------
```

If the parser had to analyse all the hypotheses given to him by this procedure, the recognition of one utterance would take several hours. But if the parser prunes acoustic hypotheses without checking whether they are grammatical or not, simply on the basis of bigram score and acoustic match, this amounts to narrowing the acoustic search beam in a severe way. The solution we choose is roughly described by the following procedure:

In each cycle, having read in the new hypotheses, the parser builds a new sorted agenda, which keeps all possible pairs of active edges and new hypos. Verification is then done in two steps.

1. The agenda is parsed until the beam threshold is reached. When a hypo is successful for the first (which is the best) time then it is verified to the decoder with the n-gram score of the predecessor word with which it could be parsed.

2. For the rest of the agenda, which is below the beam, no parsing is done any more, but it is searched for those hypos which where not verified yet. Those found are verified to the decoder with the (low) n-gram score they get from their active edge partner's predecessor word.

This leads to the effect that if a wordhypo to be verified originally was above the beam but could not be parsed, a very low n-gram score is given back. High bigram probabilities are not passed to the acoustic decoder if they correspond to unparsable word sequences. This directly influences the pruning and search behaviour of the HMM beam search, without killing paths too early in the decoder.

5 Experiments

Experiments were performed on ten test utterances in four modes: acoustic decoder without parser (AC) , incremental bottom–up interface (BU), verification interface (VE), and prediction interface (PR). The speaker dependent acoustic models were fairly well adapted and gave 90 percent word accuracy without any language model with a beam that pruned between 85 and 90 percent of all active gridpoints at each frame. The pronunciation dictionary contained 363 words. As described in the paper, the parser prunes on a score per frame basis. The parser pruning threshold in the experiments was set to 0.5 times the score per frame of the best edge spanning the utterance from the beginning to the current frame.

M	t	nBU	nTD	Utt	nE	nGP
Ac	47.4	—	—	0.5	—	—
Bu	96.9	385	—	0.7	6195	790
Ve	58.2	289	41	0.5	5827	1045
Pr	220.5	223	7621	0.7	4954	78

Figure 1: System behavior for few hypotheses

Figure 1 summarizes the results. The columns in the table contain the average time needed to recognize an utterance in seconds (t), the number of word hypotheses sent to the parser (nBU), the number of top–down messages sent from the parser to the decoder (nTD), the utterance recognition rate (Utt), the number of edges created by the parser (nE), and the maximum number of active grid points in the decoder in one Frame (nGP). The word recognition rate of the decoder alone was aproximately 0.9.

One effect of close integration was that either there was no parsable sequence of words at all, or the best scoring parsable sequence was the correct one. For that reason, the word accuracy is only given for the standalone acoustic recognition mode (AC). Five of the ten sentences were recognized correctly in all three modes BU, VE, and PR. The time and communication measurements in the table are the average values of those five utterances.

With only ten utterances for testing, the utterance recognition rates are of course of rather limited reliability. Still, it seems safe to say syntactic knowledge can be used to control the acoustic recognition process in an advantageous way.

The more important result is that the time penalty one has to pay for close integration is not as bad as expected: in verification mode, the tightly coupled system is only 20 percent slower than the HMM recognizer on its own. While it is true that there was no benefit in terms of recognition rate in verification mode, this result suggests that it should be feasible to achieve time–synchronous integration in a real time system. This may be a goal in itself for various reasons, as already mentioned in the introduction.

In BU mode, the many bottom up hypotheses slow down the parser considerably. Supplying predictions at every frame in PR mode seems to be prohibitive. Still, these two strategies increased the recognition rate, and the time consumed may be largely due to the rather straightforward implementation. Essentially the same information is computed and sent several times for each word end, so our hope is that a satisfactory speedup can be achieved in the future.

In a second experiment, we widened both the parser beam (from 0.5 to 0.3) and the acoustic beam (from $10^{-}6$ to $10^{-}8$) to see if the relative time behaviour of the different strategies would remain the same:

M	t	nBU	nTD	Utt	nE	nGP
Bu	2911.7	1465	—	0.7	19767	1712
Ve	115.8	776	135	0.7	9765	2047
Pr	282.7	415	8304	1.0	7519	184

Figure 2: System behavior for many hypotheses

We were quite surprised to see that PR mode caught up remarkably relative to VR, from a factor 3.8 to only 2.4 times slower. It seems that it is worthwhile to pursue the PR strategy if some more care is taken with efficiency considerations. The apparent breakdown in BU time performance happened because the LISP process size exceeded the main memory size and the machine began to page heavily. Depending on the parser beam and the total number of word hypotheses, there seemed to exist a critical point, where the chart begins to grow very rapidly. In BU mode, this point is reached much earlier than in the other modes.

The number of created edges gives a good impression of the work the parser has to do. With more narrow beams the differences are not too big, while with a wider beam, the parser has to perform considerably more search in BU mode.

The number of active grid points per frame reflects the amount of ambiguity the decoder has to cope with. At the first glance it seems surprising that the maximum number of active grid points is higher in VR than in BU mode. The explanation for this is that in BU and PR mode the language penalties are added when a word starts. In VR mode the penalty is added in a word's final state. Since the number of active gridpoints is always largest when a word has just begun, and exactly these gridpoints are punished by the language penalty in PR an BU mode, the number of gridpoints is larger in VR mode. However, just before a word is sent as a wordhypothesis, the language penalty shows effect. BU mode prunes more in the non final states of words. In VR mode the maximal effect of pruning takes place in word final states.

The very low nGP values in PR mode give an impression of how strongly the prediction from the parser restricts the search of the decoder. Corresponding to the number of received hypotheses, the edges created by the parser are less than in the other modes. The reason for the lower speed compared with VR is twofold. All of the submitted hypotheses must be able to lead to a further analysis, since they are a subset of the predictions given by the parser before. This might lead to long false paths which die very late. In addition to that the computation of predictions is extra work to do.

6 Conclusion

One effect of close integration of syntax and acoustics was that word error rates became essentially meaningless due to the fact that utterances were either recognized correctly or not recognized at all. This has also been observed in Goodine et al. 91 [3]. One might argue that this sacrifices flexibility compared to an approach were acoustic decoding is done like a filter *before* any linguistic analysis begins. However, measuring word accuracy in unparsable word sequences does not make sense in a speech understanding system which relies on parsing whole utterances. Rather, the conclusion should be that a syntactic module used for speech understanding must be robust with respect to recognition errors, even if it is so closely integrated with the acoustic decoder that only parsable sequences are permitted.

The experiments did not permit us to conclude that either of the two top–down strategies is superior to the other. While verify mode was much faster with narrow search beams, it resulted in less recognition rate than prediction mode. Also, the slowdown caused by prediction mode was less pronounced for wider search beams. What can be said is, that, given the pradigma of time synchronous processing, interactive strategies are superior to bottom up strategies.

The main conclusion of this paper is that it is feasible to do acoustic and syntactic analysis in a time synchronous, tightly coupled way without necessarily paying intolerable time penalties, and that it is possible to do so with the complex and powerful mechanism of a chart parser and a feature based unification grammar. The necessary restrictions did not come from either the decoder or the parser in isolation, but only from a search process guided by both knowledge sources.

References

[1] Lee-Feng Chien, Keh-Jiann Chen, and Lin-Shan Lee. A best-first language processing model integrating the unification grammar and markov language model for speech recognition applications. In *IEEE Transactions on Speech and Audio Processing*, volume 1,2, pages 221–240, 1993.

[2] Pierre Dupont. Dynamic use of syntactical knowledge in continuous speech recognition. In *Proc. Eurospeech 1993*, pages 1959–1962, 1993.

[3] David Goodine, Stephanie Seneff, Lynette Hirschman, and Michael Phillips. Full integration of speech and language understanding in the MIT spoken language system. In *Proc. Eurospeech 1991*, pages 845–848, 1991.

[4] Hynek Hermansky. Perceptual linear predictive (plp) analysis of speech. *J. Acoust. Soc. Amer.*, 87(4):1738–1752, April 1990.

[5] K. Kita, T. Kawabata, and H. Saito. Hmm continuous speech recognition using predictive lr parsing. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 703–706, 1989.

[6] Kai-Fu Lee. *Hidden Markov Modelling of Speech*, chapter 2, pages 16–43. Kluwer, 1989. in *The Development of the SPHINX system.*

[7] Hy Murveit and Robert Moore. Integrating natural language constraints into hmm-based speech recognition. In *IEEE International*

Conference on Acoustics, Speech and Signal Processing, volume 1, pages 573–576, Albuquerque, April 1990.

[8] H. Ney, R. Haeb-Umbach, et al. Improvements in beam search for 10000-word continuous speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1992.

[9] Herrmann Ney. Automatische Spracherkennung: Architektur und Suchstrategie aus statistischer Sicht. *Informatik Forschung und Entwicklung*, 7, 1992.

[10] Stuart M. Shieber. Using restriction to extend parsing algorithms for complex-feature-based formalisms. In *Proc. of the ACL 1985*, volume 23, pages 145–152, 1985.