Progress Report on the Chronus System: ATIS Benchmark Results

Roberto Pieraccini, Evelyne Tzoukermann, Zakhar Gorelov, Esther Levin, Chin-Hui Lee, Jean-Luc Gauvain

AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974

ABSTRACT

The speech understanding system we propose in this paper is based on the stochastic modeling of a sentence as a sequence of elemental units that represent its meaning. According to this paradigm, the original meaning of a sentence, can be decoded using a dynamic programming algorithm, although the small amount of training data currently available suggested the integration of the decoder with a more traditional technique. However, the advantage of this method consists in the development of a framework in which a closed training loop reduces the amount of human supervision in the design phase of the understanding component. The results reported here for the February 1992 DARPA ATIS test are extremely promising, considering the small amount of hand tuning the system required.

1. INTRODUCTION

In February 1991 [1] we proposed a novel paradigm that represents the conceptual content of a spoken sentence in terms of a probabilistic finite state automaton. The motivations for developing this model are summarized by the following points.

- Current natural language understanding systems are generally based on the synthesis of rules operated generally by an expert. This procedure makes maintenance, updating, and generalization of a system to other tasks a very expensive and difficult operation. We believe that an understanding system should incorporate a mechanism that allows, or is suitable for, unsupervised learning. Only using this mechanism can the system easily take advantage of large amounts of training data.
- Systems that are based on parsing with formal grammars (finite state, context free, etc) are generally very rigid. A system that has to be integrated with a speech recognizer has to be quite insensitive to recognition errors (substitution, insertion, deletion of words) as well as to speech disfluencies, like false starts, ungrammatical sentences, non-speech phenomena, and so on.
- The understanding model should define a framework that allows an easy and natural integration with the speech recognizer.

Following these considerations we formalized the speech understanding problem in terms of a communication channel whose input is the meaning of a sentence and whose output is a sequence of acoustic observations. Here we assume that the meaning of a sentence can be expressed by a sequence of basic meaning units $\mathbf{M} = m_1, m_2, \ldots m_{N_M}$ and that there is a sequential correspondence between each m_j and a subsequence of the acoustic observation $\mathbf{A} = a_1, a_2 \ldots a_{N_A}$. This hypothesis, although very restrictive, was successfully introduced also in [2]. According to this model of the spoken sentence production, one can think of decoding the original sequence of meaning units directly from the acoustic observation. The decoding process can be based on the maximization of the a posteriori probability $P(\mathbf{M} \mid \mathbf{A})$.

The problem now consists in defining a suitable representation of the meaning of a sentence in terms of basic units. The representation we chose was inspired by the *semantic network* [3, 4] paradigm, where the meaning of a sentence can be represented as a relational graph whose nodes belong to some category of concepts and whose arcs represent relations between concepts or linguistic cases. In our representation, each unit of meaning consists of a pair $m_j = (c_j, v_j)$, where c_j is a conceptual relation, (e.g. *origin, destination, meal* in the ATIS domain), and v_j is the value with which c_j is instantiated in the actual sentence. (e.g. *Boston, San Francisco, breakfast*). Given a certain application domain we can define two sets of symbols, $\mathcal{C}$ and $\mathcal{V}$, such that $c_j \in \mathcal{C}$, and $v_j \in \mathcal{V}$. For an application like ATIS, the size of the dictionary of concept relations $\mathcal{C}$ is fairly small (around 50), while the dictionary of concept values $\mathcal{V}$ can be relatively large (consider for instance all the possible flight numbers). Moreover, due to the limited amount of training data we may reasonably think of collecting in this task, it is advisable to have a relatively small number of parameters to be estimated. This consideration lead us to use the model for representing only the sequence of concept relations c_j. The sequence of concept values is detected using more traditional techniques by a subsequent mod-

ule called the *template generator*, that uses both the decoded concept name and the sequence of words. Hence, according to the maximum a posteriori decoding criterion, given a sequence of acoustic observations $\mathbf{A}$, we want to find a sequence of conceptual relations $\mathbf{C}$ and a sequence of words $\mathbf{W} = w_1, \ldots, w_{N_W}$ that maximize the a posteriori probability $P(\mathbf{W}, \mathbf{C} \mid \mathbf{A})$. The underlying model for computing and maximizing this probability was chosen to be a HMM whose states represent concept relations and whose observation probabilities are state-local language models in the form of word bigrams [1, 5].

2. SYSTEM ARCHITECTURE

The task of the *conceptual decoder* (see Fig. 1) is that of providing a *conceptual segmentation* $\mathbf{S} = [c_j, (w_{I_j}, w_{I_j+N_j})], j = 1, \ldots, N_M$, where $(w_{I_j}, w_{I_j+N_j}) = w_{I_j}, w_{I_j+1} \ldots w_{I_j+N_j}$ is the subsequence of words that express the concept relation c_j within the given sentence

In the current version of the CHRONUS understanding system the speech recognizer is used in a decoupled mode. The best string of words produced by the recognizer is used by the decoder for generating the conceptual segmentation. Because in this particular task there are numbers, acronyms and compound words, the string is pre-processed by a module called *lexical parser* that generates a lattice with all the possible interpretations of the string (e.g. the substring "B SEVEN FOUR SEVEN" could be interpreted as "B 747" or "B7 47" or "B74 7", etc. The conceptual decoder is then realized as a generalization of the Viterbi algorithm that works on a lattice rather than on a string of words.

The *template generator* [6] consists of a simple pattern matching procedure that, given the conceptual segmentation, produces for each concept relation c_j the corresponding concept value v_j. Finally the *SQL translator* translates the meaning representation $\mathbf{M}$ into an SQL query.

3. THE NL COMPONENT

3.1. Training the conceptual model

The conceptual model, as explained in the introduction of this paper, consists of concept transition probabilities $P(c_{K_i} \mid c_{K_{i-1}})$ and concept conditional bigram language models $P(w_i \mid w_{i-1}, c_{K_i})$, where c_{K_i} is the concept expressed by the phrase in which word w_i is included. These probabilities were initially trained using a set of 532 sentences whose conceptual segmentation was provided by hand. This initial model was used in the experiments described in [1, 5] and gave satisfactory performance as far as the conceptual segmentation of test sentences was concerned. Hand labeling training sentences is of course a rather expensive procedure whose consistence is rather doubtful. As of today, most of the training sentences available are annotated with a reference file that includes the right answer. However, for taking advantage of the annotated sentences we must use the whole understanding system in the training phase, generate the answer, and compare the answer with the reference file (see Fig. 1). Therefore the *comparator* [7] provides the training procedure with a feedback signal that can be used to partially automatize the training procedure. As a first attempt to develop a completely automatic training procedure, we designed a training loop based on the following steps:

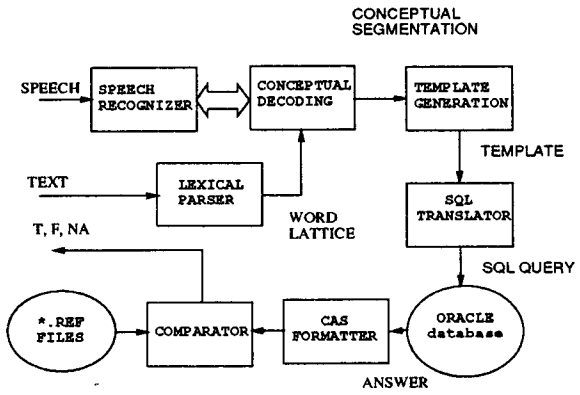

Figure 1: Block diagram of the proposed understanding system

1. Start with a reasonable model.
2. Generate an answer for each sentence in the training set.
3. Compare each answer with the corresponding reference answer.
4. Use the conceptual segmentation of the sentences that were given a correct answer to reestimate the model parameters.
5. Update the model and go to step 2

A certain number of sentences will still produce a wrong answer after several iterations of the training loop. The conceptual segmentation of these sentences may be then corrected by hand and included in the training set for a final reestimation of the model parameters. Table 1 shows the sets of data used for testing the effectiveness of the training loop. All sentences are class A (context independent) sentences and belong to the MADCOW database. The conceptual segmentation of the sentences in set A was done by hand, set B and C were annotated

Set	Number of Sentences	Description
A	532	handlabled
B	446	annotated
C	195	annotated (oct-91)

Table 1: Description of the data sets used in the training experiment

with reference files (set C corresponds to the official October 91 test set). The comparison with reference files was done using only the minimal answer. The results of this experiment are reported in Table 2. The first line in the table shows the results (as the percentage of correctly answered sentences) both on set B and on the October 91 test set when the initial model, trained on the 532 hand labeled sentences, was used. The second line shows the results on October 91 when the initial model is smoothed using the supervised smoothing described in [5]. The third line reports the accuracy (on both set B and October 91) when the sentences that were correctly answered out of set B were added to the training set (this set is called T(B)) and their conceptual labeling was used along with set A for reestimating the model. It is interesting to notice that the performance on the October 91 test set is higher than that obtained with supervised smoothing. The last line of Table 2 shows that supervised smoothing increases the performance by a very small percentage. The results of this experiment show that the use of automatically produced conceptual segmentation along with the feedback introduced by the comparator improves the performance of the system of an amount that is comparable with that obtained by a supervised procedure, like the supervised smoothing.

3.2. The dialog manager

For dealing with class D sentences we developed a module, within the template generator, called the *dialog manager*. The function of this module is to keep the history of the dialog. In this version of the dialog manager the history is kept by saving the template from the previous sentence in the same session and merging it with the newly formed template, according to a set of application specific rules.

Training set	% correct on set B	% correct on set C
A	48.2	63.5
A+smooth		72.3
A+T(B)	50.9	72.8
A+T(B)+smooth		73.3

Table 2: Results using the training loop described in the text. T(B) is the subset of B that was correctly answered by the system.

Class	#	T	F	NA	W. Err.
A	402	256	96	50	60.2
D	285	122	113	50	96.8
A+D	687	378	209	100	75.4

Table 3: Official NIST score for the NL ATIS February 92 test

3.3. NL results on February 1992 test

The February 1992 test set includes 402 class A sentences and 285 class D sentences. This set of 687 sentences, used for scoring the NL performance, is part of a larger set that originally included 283 class X (unanswerable) sentences. The test was carried out for the overall set of 970 sentence, without knowing which class they belong to. The official score given from NIST is summarized in Table 3. After the test we found an inaccuracy in the module of the SQL translator that is responsible for the CAS formatting. We fixed the bug and rescored the whole set of sentences, obtaining the results reported in Table 4. In Table 5 we report a detailed analysis of the results. In this analysis we included only the sentences that generated a false response. Conceptual decoding and template generator errors are generally due to the lack of training data. SQL translator and dialog manager errors are generally due to the limited power of the representation we are currently using. Finally for the errors attributed to the CAS format or labeled as ambiguos we generated a correct internal meaning representation but the format of the answer did not comply with the principles of interpretation, or our interpretation did not agree with the one given by the annotators.

4. THE SPEECH RECOGNIZER

In this section we give a description of the speech recognition system that was used in conjunction with the natural language understanding system for the February 92 ATIS test. Other details can be found in [8, 9]

Class	#	T	F	NA	W. Err
A	402	299	54	49	39.0
D	285	167	67	51	64.9
A+D	687	466	121	100	49.8

Table 4: Score for the NL ATIS February 92 after the format bug was removed

Error type	Number of Sentences
Conceptual decoding	30
Template generation	19
SQL translator	24
CAS format	16
Dialog manager	20
Ambiguous	12

Table 5: Analysis of the errors for the NL ATIS February 92 test

Data origin	# of utterances	word error	string error
MIT	193	9.7	47.2
BBN	194	13.1	58.8
CMU	193	17.8	75.1
SRI	193	21.5	68.4
ATT	197	28.3	76.1
OVERALL	970	17.5	64.6

Table 6: Score for the SPREC ATIS February 92 test

The Speech signal was first filtered from 100 Hz to 3.8 KHz and down-sampled to an 8 kHz sampling rate. 10th order LPC analysis was then performed every 10 msec on consecutive 30 msec windows with a 20 msec frame overlap. Based on the short-time LPC features, 12 LPC-derived cepstral coefficients and their first and second derivatives, plus normalized log energy and its first and second derivatives were computed and concatenated to form a single 39-dimension feature vector.
6259 spontaneous utterances from the MADCOW data were used for training the acoustic models. Context-dependent phone-like units [10], including double-context phones, left-context phones, right-context phones, context-independent phones, word-juncture context dependent phones and position dependent phones, were modeled using continuous density hidden Markov models (HMM) with mixture Gaussian state observation densities. The inventory of acoustic units was determined through an occurrency selection rule. Only units that appear in the training database more than 20 times were selected, resulting in a set of 2330 context-dependent phones. A maximum of 16 mixture components was used for each acoustic HMM state. The HMM parameters were estimated by means of the segmental k-means training procedure [11].
The recognition lexicon consisted of 1153 lexical entries including 1060 words appearing in the Feb91 benchmark evaluation and 93 compound words which were mostly concatenation of letters to form acronyms. Each entry had a single pronunciation. In addition, two non-phonetic units, one for modeling weak extraneous (out of vocabulary) speech events and the other for modeling strong extraneous speech events, were included, like in [12].
Word bigrams were used in the test. They were estimated using the same set of 6259 annotated sentences, and smoothed with backoff probabilities. The perplexity of the language defined by the bigram probabilities, computed on the training set, was found to be 17.

4.1. SPREC results on February 1992 test

The speech recognition results are summaried in Table 6 Overall we observed 17.5% word error and 64.6% string error.

In the current system configuration, only 6259 utterances (about 12 hours of speech) were used to create the acoustic HMM models. Out of the 218 speakers, 15 of them were from the ATT training set and 17 of them were from the CMU training set, which amounts to about 90 minutes of training data from each of them. We can see from Table 6 that there is a problem due to an insufficient training for ATT and CMU test data. On the other hand, since most of the training data we used were collected at BBN and MIT, the performance is better for BBN and MIT test speakers.
94 out of the 427 deleted words were A and THE. Short function words amounted to over 90% of the deletion errors. As for the 328 insertion errors, 46 of them were insertion of words A and THE. Again, short function words contributed to over 90% of the insertion errors. Since function words, in most cases, did not affect the meaning of a recognized sentence, we expect that such errors did not degrade the performance of the NL module.
Substitution errors had a greater impact on the SLS system performance than insertion and deletion errors. Most of the substitution errors can be categorized into three types:

1. Out-of-vocabulary words;
2. Morphological inflections of words, which are difficult to discriminate acoustically for band-limited data;
3. short function words.

Out of the 1153 substitution error, 66 were caused by out-of-vocabulary words, and 127 were caused by morphological inflections. For the remaining 85% of the er-

Class	#	T	F	NA	W. Err.
A	402	208	118	76	77.6
D	285	92	115	78	108.1
A+D	687	300	233	154	90.2

Table 7: Official NIST score for the SLS ATIS February 92 test

rors, about half involved short function words.

5. SLS RESULTS ON FEBRUARY 1992 TEST

The integrated SLS system for the February 1992 test was implemented by using the best first recognized string from our speech recognizer as input to the NL system. Table 7 reports the official results from NIST and Table 8 reports our results after the format bug was fixed.

6. CONCLUSIONS

In this paper we give a global outline of the CHRONUS speech understanding system. The system is built around the conceptual decoder, a Viterbi decoder that uses a stochastic model for extracting the conceptual content of an input sentence. Although the problem is formalized in such a way that the decoder could also extract the actual value of the conceptual relations (not only their category), the limited amount of training sentences currently available suggested the use of a more traditional pattern matcher (the template generator) along with the conceptual decoder. However, the advantage of the stochastic formalization is the trainability of the model over a database of suitably annotated examples. The annotation given with the MADCOW sentences and the comparator developed by NIST provide a useful feedback signal that allows to automatize the training procedure. In a preliminary experiment designed to test this procedure we show that a significant improvement of the accuracy of the system can be obtained without human supervision.
The results on the February 92 ATIS test are then reported in the paper. The big discrepancy between the official NIST score and the score obtained in a successive assessment of the system is explained by inaccuracies found in the answer formatter, that we don't believe affects the *real* performance of the CHRONUS system. Nevertheless, this suggests the importance of investigating a more meaningful and more rubust scoring criterion.

Class	#	T	F	NA	W. Err.
A	402	237	89	76	63.2
D	285	117	90	78	90.5
A+D	687	354	179	154	74.5

Table 8: Score for the SLS ATIS February 92 after the format bug was removed

REFERENCES

1. Pieraccini, R., Levin, E., Lee, C. H., "Stochastic Representation of Conceptual Structure in the ATIS Task," *Proc. of 4th DARPA Workshop on Speech and Natural Language*, Asilomar (CA), February 1991.
2. Prieto, N., Vidal, E., "Learning Language Models through the ECGI Method," *Proc. of EUROSPEECH 91*, Genova, Italy, September 1991.
3. Simmons, R. *Semantic networks: their computation and use for understanding English sentences*, In Schank and Colby, eds Computer Models of Thought and Language, Freeman: San Francisco, 1973.
4. Kittredge, *Analyzing language in restricted domains: Sublanguage description and processing*, In Grishman, ed. Lawrence Erlbaum, 1986.
5. Pieraccini, R., Levin, E., "Stochastic Representation of Semantic Structure for Speech Understanding," *Proc. of EUROSPEECH 91*, Genova, Italy, September 1991.
6. Pieraccini, R., Tzoukermann, E., Gorelov, Z., Gauvain, J. L., Levin, E., Lee, C. H., Wilpon, J. G., "A Speech Understanding System Based on Statistical Representation of Semantics," *Proc. of ICASSP 92*, San Francisco, CA, March 1992.
7. Bosen, S., Ramshaw, L., Ayuso, D., Bates, M., "A Proposal for SLS Evaluation," *Proc. of 2nd DARPA Workshop on Speech and Natural Language*, Cape Cod (MA), October 1989.
8. Lee, C. H., Giachin, E., Rabiner, L. R., Pieraccini, R., Rosenberg, A. E., "Improved Acoustic Modeling for Speaker Independent Large Vocabulary Continuous Speech Recognition," *Proc. of ICASSP 1991*, Toronto, Ontario, May 1991.
9. Gauvain, J. L., Lee, C. H., "Bayesian Learning fro Hidden Markov Models with Gaussian Mixture State Observation densities," *Proc. of EUROSPEECH 91*, Genova, Italy, September 1991.
10. Lee, C. H., Rabiner, L. R., Pieraccini, R., Wilpon, J. G., "Acoustic Modeling for Large Vocabulary Speech Recognition," *Computer, Speech and Language*, 4, pp. 127-165, 1990.
11. Rabiner, L. R., , Wilpon, J. G., Juang, B. H., "A segmental k-means training procedure for connected word recognition based on whole word reference patterns," *AT&T Technical Journal*, vol. 65 no. 3, pp 21-31, May/June 1986
12. Wilpon, J. G., Rabiner, L. R., Lee, C. H., Goldman, E. R., "Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models," *IEEE Trans. ASSP*, Vol. 38, No. 11, pp.1870-1878