A Speech Understanding System Based on Statistical Representation of Semantics

Roberto Pieraccini Evelyne Tzoukermann Zakhar Gorelov Jean-Luc Gauvain
Esther Levin Chin-Hui Lee Jay G. Wilpon

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

ABSTRACT

An understanding system, designed for both speech and text input, has been implemented based on statistical representation of task specific semantic knowledge. The core of the system is the conceptual decoder, that extracts the words and their association to the conceptual structure of the task directly from the acoustic signal. The conceptual information, that is also used to disambiguate the English sentences, is encoded following a statistical paradigm. A template generator and an SQL translator process the sentence and produce SQL code for querying a relational database. Results of the system on the official DARPA test are given.

1. INTRODUCTION

The goal of a speech understanding system is to translate a sequence of acoustic measurements of a speech signal into some form that represents the meaning conveyed by the sentence. One of the knowledge representation paradigms, known as *semantic networks* [2], establishes relations between conceptual entities through a graph structure. These *concept relations*, or *linguistic cases*, can be used to label the phrases of a sentence and obtain an intermediate representation useful for its interpretation. In a limited domain task [1] (e.g. airline reservation, database retrieval, etc.) the number of different concepts can be assumed to be finite and directly deducible from the knowledge of the task itself. For example, a typical query in an airline reservation domain is the following:

SHOW ME ALL THE NONSTOP FLIGHTS FROM DALLAS TO DENVER LEAVING ON APRIL TWENTY SECOND.

A non conventional approach that incorporates the conceptual model into the acoustic decoder has been taken. Therefore, the speech recogniser will decode the words and their association to the concepts directly from the acoustic signal. The output of this module is called *conceptual segmentation*; Table 1 shows an example of conceptual segmentation: A second module (the *template generator*) translates the initial segmentation into a different template conforming to a more abstract formalism. Finally an *SQL translator* generates SQL query for extracting the requested information from an Oracle database. Fig. 1 shows a block diagram

QUERY:	*SHOW ME ALL*
STOP-NUMBER:	*THE NONSTOP*
QUERY-OBJECT:	*FLIGHTS*
FLIGHT-ORIGIN:	*FROM DALLAS*
FLIGHT-DESTINATION:	*TO DENVER*
FLIGHT-DATE:	*LEAVING ON APRIL TWENTY SECOND*

Table 1: Conceptual Segmentation

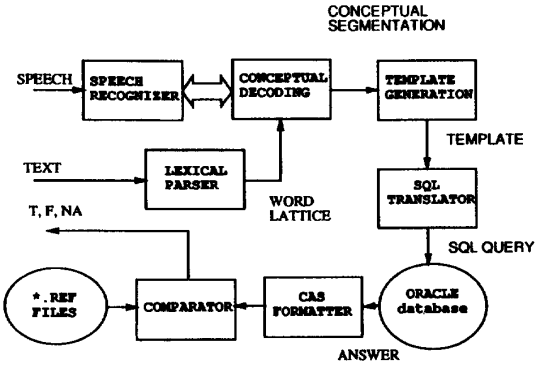

Figure 1: Block diagram of the proposed understanding system

of the whole system. Once the answer for a sentence is produced, it is formatted according to rules defined within the DARPA ATIS project [7] and compared with a reference answer, provided with each test sentence, by a *comparator* developed by the National Institute of Standards and Technology [6]. The output of the comparator is `TRUE`, in case of exact match of the hypothesised answer and the reference answer, `FALSE`, in case of mismatch, and `NO_ANSWER` when the system cannot answer the query.

0-7803-0532-9/92 $3.00 © 1992 IEEE

2. THE CONCEPTUAL DECODING

Once the finite set of labels has been decided, the task of giving the appropriate concept label to each phrase of a sentence is performed by a statistical technique. Let us assume a spoken sentence is represented by a sequence of acoustic observations:

$$\mathbf{A} = a_1, a_2 \ldots a_N \tag{1}$$

The sentence corresponds to a sequence of words:

$$\mathbf{W} = w_1, w_2 \ldots w_M \tag{2}$$

and each word can be associated with a concept label, hence:

$$\mathbf{C} = c_1, c_2 \ldots c_M \tag{3}$$

The goal is to detect W and C given that A was observed. The problem can be approached using the *maximum a posteriori* decoding criterion, according to which we want to find the maximum of the conditional probability of W and C given A,

$$\tilde{\mathbf{W}}, \tilde{\mathbf{C}} : P(\tilde{\mathbf{W}}, \tilde{\mathbf{C}} \mid \mathbf{A}) = \max_{\mathbf{W} \times \mathbf{C}} \mathbf{P}(\mathbf{W}, \mathbf{C} \mid \mathbf{A}) \tag{4}$$

This conditional probability can be written, using the Bayes inversion formula, as:

$$P(\mathbf{W}, \mathbf{C} \mid \mathbf{A}) = \frac{\mathbf{P(A \mid W, C)P(W \mid C)P(C)}}{\mathbf{P(A)}} \tag{5}$$

The three terms on the right hand side of (5) are the acoustic model of words, the *concept-conditional* language model, and the *conceptual* model respectively. The acoustic model of a word can be reasonably assumed to be independent of the concept it expresses (i.e. the same word expressing different concepts has the same acoustic representation, hence $P(\mathbf{A} \mid \mathbf{W}, \mathbf{C}) = \mathbf{P(A \mid W)}$), and can be implemented with the standard methods (e.g. HMMs of phonetic sub-word units [8]). Making some reasonable simplifying assumptions, we chose to implement the concept-conditional and the conceptual models by a HMM whose states represent concept relations and whose observation probabilities constitute state-local language models in the form of bigrams of words [3, 4].

The conceptual stochastic segmentation

This paradigm, called CHRONUS (Conceptual Hidden Representation of Natural Unconstrained Speech), used a model of 47 states and a set of 547 training sentences that were initially hand-segmented into concepts. The tested system (the *conceptual decoder*), accepts input text sentences belonging to the domain, and produces a segmentation of them into conceptual constituents, like the one shown in Table 1. The parameters of the models were then estimated with a few iterations of the Viterbi training algorithm. The test was performed on a different set of sentences, comparing the segmentation obtained by the conceptual decoder with manual segmentation. The results on the official DARPA June 1990 and February 1991 DARPA test sets gave high segmentation accuracy (95.0% and 94.1% concepts were correctly segmented and labeled in the two test sets respectively [3]). Although the result of this test was encouraging, it did not give us an accurate estimation of the performance of the full understanding system. The generation of the correct answers from the conceptual segmentation of a sentence requires further processing, and performance of subsequent stages can affect the accuracy of the system.

The Lexicon

To relieve the problem of parameter estimation it is convenient to pay attention to the representation of lexicon. In particular, it is useful to create word classes, hence having word class bigrams instead of word bigrams in the conceptual representation. The word classes should be broad enough to allow a more robust estimation of the conceptual model parameters, but still semantically meaningful to allow the decoding of the concepts. In many cases word classes will help concept generalization. For instance in phrases like *GOING FROM BOSTON, GOING FROM ATLANTA*, etc., it is better to have a model representing the structure *GOING TO <city_name>* . Thus, we assumed the following in the design of the lexicon:

- Words with the same base (i.e. morphological variants of the same word) are grouped together. According to this principle, words like *GO, GOES, GOING* are represented by the same *super-word GO(ES)(ING).*
- Articles (*A and* THE), unevenly used in spontaneous language, are in general associated with the following word to form a single item (e.g. *THE FLIGHT →[THE]FLIGHT).* Then, for instance, the bigram *OF FLIGHT* will have the same probability as the bigrams *OF [THE]FLIGHT* and *OF [A]FLIGHT.*
- Some common compound phrases are converted into hyphenated compound expressions, like for instance *ONE WAY, ONE-WAY*, etc.
- Acronyms and numbers are represented by regular grammars, and the grammars are considered as classes of words (e.g. *TWA, USAIR*, etc.).
- Obvious semantically meaningful classes of words are grouped together (e.g. city names, aircraft names, etc.). The word classes are usually represented by finite state automatons, since they can include also sequences of words (e.g. *SAN FRANCISCO, DALLAS FORT WORTH*, etc.).
- For a given concept there are words that, still carrying different or slightly different meanings, can be grouped together according to their use in the phrases. For instance, for the concept **ORIGIN**, the words

  DEPART(S) LEAVE(S) ARRIVE(S)

  can be considered as synonyms, and can be interchanged in sentences such as:

THE FLIGHT THAT DEPART(S) FROM DALLAS
THE FLIGHT THAT LEAVE(S) FROM DALLAS
THE FLIGHT THAT ARRIVE(S) FROM DALLAS

Although the word *ARRIVE(S)* is not synonym of *DEPART(S)* and *LEAVE(S)*, in the presence of the preposition *FROM* it conveys a similar information in this context. A number of groups of synonyms were manually detected for each concept. The occurence frequencies inside a group were equally shared among the constituting words, giving the same bigram probability for synonymous words.

For a correct interpretation of the query, the conceptual decoding must take into account all the possible lexical interpretations. Hence, given an input text sentence, a lattice is generated which includes all the possible interpretations [4].

Integration with a speech recognizer

The design of a speech recogniser that maximises equation in (5) poses several implementation issues, the most important of which is the increased search space. In fact each word of the vocabulary must be represented by a model in each concept state, increasing the search space of a factor equal to the number of concepts. Since the vocabulary is of the order of 1000 words in this application and there are about 50 concepts, the speech recognition system has an active vocabulary of 50000 words. In a first implementation we limited the vocabulary of each concept only to the words that were conditionally associated to that concept. in the training set. With a training set of 547 sentences, the overall active vocabulary consisted of 2372 words, while the lexicon was composed of only 506 different words. Additionally, for making the system more simple, we also approximated the finite state grammars inside each concept (e.g. numbers, acronyms, and compound words) with word bigrams. Of course these approximations make the performance of the overall system depend heavily on the size of the training set. We tried also to use a decoupled approach [5]. In this case the speech is decoded using concept independent bigrams that were estimated on a corpus of 10,000 sentences. The resulting active vocabulary was, in this case, composed of 1153 words. We compared the two approaches (i.e. integrated and decoupled) on the basis of the word accuracy at the speech recognition level.

The experiment was carried out using a set of 47 context independent phonetic units (each one represented by a 3-state, mixture-density HMM) trained on a set of 3460 sentences. Table 2 gives the word accuracy, sentence accuracy, deletion and insertion percentages of the two recognizers on the June 1991 DARPA test, consisting of 93 sentences. In principle, the test set perplexity of the CHRONUS model (i.e. context dependent bigrams) should be lower if the model parameters are estimated on the same amount of data as the bigram model. Since the CHRONUS model was estimated on a 20 times smaller set than the conceptual-independent bigram model, the resulting test set perplexity is larger and the corresponding recognition results are worse. Hence, increasing the size of the training data for the chronus model should improve the performance of the integrated system. The question is that we still do not know which size of the overall active vocabulary in the chronus model is, that will allow such an improvement in the performance. If it is too big, the choice has to go in favor of a decoupled approach, but in this case we must adopt a *N-best* interface [9, 10, 11] or a word lattice approach [12]. Increasing the size of the training set requires more annotated data. Hand-labeling training sentences is an expensive and slow process. The comparator can be used for speeding up this procedure. If reference answers are provided, the training sentences can be used as a text input to the understanding system. The sentences whose hypothesized answer matches the reference answer are collected; their conceptual segmentation, obtained through an initial model, is used for further training of the system. The sentences that provide a wrong answer are then manually segmented. We hope that this procedure will allow to increasingly reduce the time needed for handlabeling new training sentences.

	Del	Ins	W. Acc.	S. Acc
integrated	7.3	5.3	72.3	15.2
decoupled	2.6	3.1	82.1	26.1

Table 2: Speech recognition performance for the integrated and decoupled systems

3. THE TEMPLATE GENERATOR

The function of the template generator is to translate the conceptual representation of the sentence into a template representation. A template is a table that includes a number of pairs (**TOKEN**, **value**), where **TOKEN** belongs to a finite dictionary of token names and **value** belongs to a finite (or infinite, e.g. for numeric values) dictionary of possible values that the specific token can assume. The concept name in the conceptual segmentation translates directly to the token name according to a translation table (e.g. **QUESTION→QUERY**), or according to an inference mechanism that induces information from the English noun phrase associated to the concept. In the example such as:

FLIGHT-ORIGIN : *FLY FROM DENVER*

the **FLIGHT-ORIGIN** concept is translated to the **ORIGIN-CITY** token. If, instead of *DENVER*, it were *SAN FRANCISCO INTERNATIONAL*, the returned token would be **ORIGIN-AIRPORT**.

The determination of the token value is generally more complicated and requires a set of concept-specific rules.

Table 3 shows an example of template generation corresponding to the conceptual segmentation of Table 1: The tokens have been classified according to the values they can assume. In categories such as **FARE** or **QUERY**, a specific operator needs to be returned. In Table 3 example, the token **QUERY** is given the value **LIST** which is the action required by the sentence. A pattern-matching mechanism

QUERY:	LIST
OBJECT:	flight
STOPS:	0
ORIGIN-AIRPORT:	DFW
DEST-AIRPORT:	DEN
DAY-NAME:	SUNDAY

Table 3: Example of Template

retrieves the proper object, which is a database attribute. Since the object of the input sentence is *flights*, the information that has to be given to the user is the flight identification. In categories such as **MEAL** or **STOP**, a logical value will be returned. The word *NONSTOP* is translated into the value 0 for the token **STOPS**. In categories such as **ORIGIN-AIRPORT** or **DEST-AIRPORT**, the value, to be returned is the database value which is the airport code. The codes DFW and DEN correspond to the words *DALLAS* and *DENVER*. The categories that deal with **DATE** or **TIME** require the use of a grammar to parse the relevant information. The phrase *LEAVING ON APRIL TWENTY SECOND* is translated into the corresponding day of the week (SUNDAY) needed for retrieving the appropriate flights.

The template generator could be, at least in principle, included in the stochastic framework expressed by equation (4). We decided to keep it as a separate module, because the small number of training samples we have available at the moment will make the estimation of a more complicated model not reliable enough.

4. THE SQL TRANSLATOR

The last part of the interpretation process, namely access to the required information, is implemented through a translator that dynamically generates the SQL query in order to retrieve the data. Once the template is produced by the generator, it is preprocessed and then each template is treated according to the **OBJECT**, e.g. *flight, fare, meal*, etc. Each token inside the template is interpreted in relation to its table; for example DFW, the value of **ORIGIN-AIRPORT**, or DEN, the value of **DEST-AIRPORT**, will combine to create the origin and the destination of the flight table. When the token is not directly related to the table, a link function is invoked in order to perform the join between the template object and another table object. This is the case, for example, for queries that deal with flight and fare information.

5. RESULTS AND CONCLUSION

As shown in this paper,the integration of a conceptual model in an acoustic decoder of a speech recognizer reduces the search of the recognition process. The system was tested on text understanding and the results are encouraging. Out of 195 test sentences and according to the official DARPA answers for the ATIS task, the improved system correctly answers 141 context-independent queries, which is over 72% success rate. With speech input in the decoupled approach, the results on the same test set gave more than 50% success rate. More complex sentences (e.g. queries with multiple flight-identifications, origins, and destinations) require the complexity of the conceptual model to be increased. Some modifications are also necessary for the system to handle context-dependent queries. The system should improve with the addition of training data and the elaboration of a more sophisticated model.

REFERENCES

[1] Kittredge, *Analyzing language in restricted domains: Sublanguage description and processing*, In Grishman, ed. Lawrence Erlbaum, 1986.

[2] 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.

[3] 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.

[4] Pieraccini, R., Levin, E., "Stochastic Representation of Semantic Structure for Speech Understanding," *Proc. of EUROSPEECH 91*, Genova, Italy, September 1991.

[5] Pieraccini, R., Lee, C. H., "Factorization of Language Constraints in Speech Recognition," *Proc. 29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, CA, June 1991.

[6] Boisen, S., Ramshaw, L., Ayuso, D., Bates, M., "A Proposal for SLS Evaluation," *Proc. of 2nd DARPA Workshop on Speech and Natural Language*, pp. 135-146, Cape Code (MA), October 1989.

[7] Price, P. J., "Evaluation of Spoken Language Systems: the ATIS Domain," *Proc. of 3rd DARPA Workshop on Speech and Natural Language*, pp. 91-95, Hidden Valley (PA), June 1990.

[8] Lee, C. H., Rabiner, L. R., Pieraccini, R., Wilpon, J. G., "Acoustic Modeling for Large Speech Recognition," *Computer, Speech and Language*, 4, pp. 127-165, 1990.

[9] Kubala, F., Austin, S., Barry, C., Makhoul, J., Placeway, P., Schwartz, R., "Byblos Speech Recognition Benchmark Results,", *Proc. of 4th DARPA Workshop on Speech and Natural Language*, Asilomar (CA), February 1991.

[10] Ostendorf, M., Kannan, A., Austin, S., Kimball, O., Schwartz, R., Rohlicek, J. R., "Integration of Diverse Recognition Methodologies Through Reevaluation of N-Best Sentence Hypotheses," *Proc. of 4th DARPA Workshop on Speech and Natural Language*, Asilomar (CA), February 1991.

[11] Zue, V., Glass, J., Goodline, D., Leung, H., McCandless, M., Phillips, M., Polifroni, J., Seneff, S., "Recent Progress on the VOYAGER System," *Proc. of 3rd DARPA Workshop on Speech and Natural Language*, pp. 91-95, Hidden Valley (PA), June 1990.

[12] Baggia, P., Fissore, L., Gerbino, E., Giachin, E., Rullent, C., "Improving Speech Understanding Performance through Feedback Verification," *Proc. of EUROSPEECH 91*, Genova, Italy, September 1991.