A Learning Approach to Natural Language Understanding

Roberto Pieraccini *Esther Levin*

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

Abstract

In this paper we propose a learning paradigm for the problem of understanding spoken language. The basis of the work is in a formalization of the understanding problem as a communication problem. This results in the definition of a stochastic model of the production of speech or text starting from the meaning of a sentence. The resulting understanding algorithm consists in a Viterbi maximization procedure, analogous to that commonly used for recognizing speech. The algorithm was implemented for building a module, called *conceptual decoder* for the decoding of the conceptual content of sentences in an airline information domain. The decoding module is the basis on which a complete prototypical understanding system was implemented and whose performance are discussed in the paper. The problems, the possible solutions and the future directions of the learning approach to language understanding are also discussed in this paper.

1 Introduction

A natural language understanding system is a machine that produces an action as the result of an input sentence (speech or text). There are examples [14] of systems that are able of modeling and learning the relationship between the input sentence and the action in a direct way. However, when the task is rather complicated, i.e. the set of possible actions is extremely large, we believe that it is necessary to rely on a intermediate symbolic representation. Fig. 1 depicts a natural language understanding as composed of two components. The first, called *semantic translator* analyzes the input sentence in natural language *(N-L)* and generates a representation of its meaning in a formal semantic language *(S-L)*. The *action transducer* converts the meaning representation into statements of a given computer language *(C-L)* for executing the required action.

Although there are several and well established ways [5] of performing the semantic translation with relatively good performance, we are interested in investigating the possibility of building a machine that can learn how to do it from the observation of examples. Traditional *non-learning* methods are based on grammars (i.e. set of rules) both at the syntactic and semantic level. Those grammars are generally designed by hand. Often the grammar designers

rely on corpora of examples for devising the rules of a given application. But the variety of expressions that are present in a language, even though it is restricted to a very specific semantic domain, makes the task of refining a given set of rules an endless job. Any additional set of examples may lead to the introduction of new rules, and while the rate of growth of the number of rules decreases with the number of examples, larger and larger amounts of data must be analyzed for increasing the coverage of a system. Moreover, if a new different application has to be designed, very little of the work previously done can be generally exploited.

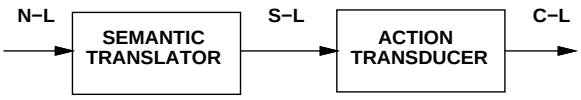

Figure 1: Understanding as a translation process

The situation is even more critical for spoken rather than written language. Written language generally follows *standard* grammatical rules more strictly than spoken language, that is often ungrammatical and idiomatic. Besides, in spoken language, there are phenomena like false starts and broken sentences that do not appear in written language. The following is a real example from a corpus of dialogues [21] within the airline information domain, (DARPA ATIS project [13], see section 3).

> *FROM uh sss FROM THE PHILADELPHIA AIRPORT um AT ooh THE AIRLINE IS UNITED AIRLINES AND IT IS FLIGHT NUMBER ONE NINETY FOUR ONCE THAT ONE LANDS I NEED GROUND TRANSPORTATION TO uh BROAD STREET IN PHILELD PHILADELPHIA WHAT CAN YOU ARRANGE FOR THAT* [1]

It is clear from this example that rules for analyzing spontaneously spoken sentences can hardly be foreseen by a grammarian. We believe that a system that learns from examples will ease the work of a designer of text or speech understanding system giving the possibility of analyzing big corpora of sentences. The question remains on how to collect those corpora, which kind of annotation is needed, and what is the amount of manual work that has to be carried on.

The basis of the work exposed in this paper is a semantic translator, called *CHRONUS.* [2] CHRONUS is based on a stochastic representation of conceptual entities resulting from the formalization of the speech/text understanding problem as a communication problem. The paper is structured as follows. In section 2 we formalize the language understanding problem and we propose an algorithm based on the maximum a posteriori decoding. In section 3 we explain how the described algorithm for conceptual decoding can be part of a complete understanding system and we give a short description of all the modules that were implemented for an information retrieval application. In section 4 we discuss experimental performance of the system as well as issues related to the training of the conceptual decoder. Finally in section 5

[1] this sentence was cited by Victor Zue, MIT, during the 5th DARPA Workshop on Speech and Natural Language, Harryman, NY, Dec. 1991.

[2] *CHRONUS* stands for Conceptual Hidden Representation of Natural Unconstrained Speech.

SHOW ME THE FLIGHTS TO BOSTON	(question,display) (subject,flight) (destin,BBOS)
HOW MUCH IS THE PRICE OF THE FLIGHT FROM ATLANTA	(question,display) (subject,fare) (destin,MATL)
IS BREAKFAST SERVED ON THE FLIGHT?	(question,yes-no) (subject,breakfast)

Table 1: Example of keyword/pair representations of simple phrases within the ATIS domain.

we conclude the paper with a discussion on the open problems and the future developments of the proposed learning paradigm.

2 Formalization of the Language Understanding Problem

In this section we propose a formalization of the language understanding problem in terms of the noisy channel paradigm. This paradigm has been introduced for formalizing the general speech recognition problem [2] and constitutes a basis for most of the current working speech recognizers. Recently, a version of the paradigm was introduced for formalizing the problem of automatic translation between two languages [8]. The problem of translating between two languages has the same flavor of the problem of understanding a language [19]. In the former, both the input and the output are natural languages, while in the latter the output language is a formal semantic language apt to represent meaning.

The first assumption we make is that the meaning of a sentence can be expressed by a sequence of basic units $\mathcal{M} = \mu_1, \mu_2, \ldots, \mu_{N_M}$ and that there is a *sequential correspondence* between each μ_j and a subsequence of the acoustic observation $\mathbf{A} = a_1, a_2 \ldots a_{N_A}$, so that we could actually segment the acoustic signal into consecutive portions, each one of them corresponding to a phrase that express a particular μ_i. The second assumption consists in thinking of the acoustic representation of an utterance as a version of the original sequence of meaning units corrupted by a noisy channel whose characteristics are generally unknown. Thus, the problem of understanding a sentence can be expressed in this terms: given that we observed a sequence of acoustic measurements $\mathbf{A}$ we want to find which semantic message $\mathcal{M}$ most likely produced it, namely the one for which the a posteriori probability $P(\mathcal{M} \mid \mathbf{A})$ is maximum. Hence the problem of understanding a sentence is reduced to that of maximum a posteriori probability decoding (MAP).

For the actual implementation of this idea we need to represent the meaning of a sentence as a sequence of basic units. A simple choice consists in defining a unit of meaning as a *keyword/value* pair $m_j = (k_j, v_j)$, where k_j, is a conceptual category (i.e. a *concept* like for instance *origin of a flight, destination, meal*) and v_j is the *value* with which k_j is instantiated in the actual sentence (e.g. *Boston, San Francisco, breakfast*). Given a certain application domain we can define a concept dictionary Γ and for each concept $\gamma_j \in \Gamma$ we can define a set of values $\Upsilon^j = \{\varphi 1^j, \varphi 2^j, \ldots, \varphi N_v^j\}$. Examples of meaning representation for phrases in the airline information domain are given in Table 1.

For information retrieval applications, the number of concepts Γ is relatively small (about

50, in the ATIS application), while the dictionary of concept values Υ can be relatively large (consider for instance all the possible flight numbers in the airline reservation domain). Moreover, the limited amount of training data available at the time we developed the system (a few thousand sentences) suggested to keep the number of model parameters relatively small.

Therefore, in the decoding process, we considered only concepts $k_j \in \Gamma$; the concept values $v_j \in \Upsilon$ are derived through pattern matching functions at a later stage of processing.

In the remaining of the section we will use the following notations: $\mathbf{W} = w_1, \ldots, w_{N_W}$ is the sequence of words actually uttered in the sentence represented by the acoustic observation $\mathbf{A}$ and $\mathbf{C} = c_1, \ldots, c_{N_W}, \quad c_i \in \Gamma$ is the corresponding sequence of concepts labels. A consecutive set of words $w_i, \ldots, w_{i+k}$ labeled by the same concept label, constitutes a phrase expressing a concept, thus defining a segmentation of the sentence into concepts, referred to, in the following, as *conceptual segmentation*.

Hence, according to the maximum a posteriori decoding criterion, given the sequence of acoustic observations $\mathbf{A}$, we want to find the sequence of conceptual labels $\tilde{\mathbf{C}}$ and the sequence of words $\tilde{\mathbf{W}}$ that maximize the a posteriori probability $P(\mathbf{W}, \mathbf{C} \mid \mathbf{A})$, namely

$$P(\tilde{\mathbf{W}}, \tilde{\mathbf{C}} \mid \mathbf{A}) \;=\; \max_{\mathbf{W},\mathbf{C}} \; P(\mathbf{W}, \mathbf{C} \mid \mathbf{A}). \tag{1}$$

Using the Bayes inversion formula, the conditional probability can be rewritten as:

$$P(\mathbf{W}, \mathbf{C} \mid \mathbf{A}) \;=\; \frac{P(\mathbf{A} \mid \mathbf{W}, \mathbf{C}) P(\mathbf{W} \mid \mathbf{C}) P(\mathbf{C})}{P(\mathbf{A})}, \tag{2}$$

and since $P(\mathbf{A})$ is constant:

$$\arg\max_{\mathbf{W},\mathbf{C}} \; P(\mathbf{W}, \mathbf{C} \mid \mathbf{A}) \;=\; \arg\max_{\mathbf{W},\mathbf{C}} \; P(\mathbf{A} \mid \mathbf{W}, \mathbf{C}) P(\mathbf{W} \mid \mathbf{C}) P(\mathbf{C}) \tag{3}$$

In this formula $P(\mathbf{C})$ represents the a-priori probability of the sequence of concepts, $P(\mathbf{W} \mid \mathbf{C})$ is the probability of the sentence (intended as a sequence of words), given that the sentence conveys the sequence of concepts $\mathbf{C}$, and $P(\mathbf{A} \mid \mathbf{W}, \mathbf{C})$ is the acoustic model. The three components of the conditional probability in 2 can be thought of also as a semantic (probability of meanings), syntactic (probability of words given a meaning), and acoustic component respectively. We can then reasonably assume that the acoustic representation of a word is independent from the conceptual relation it belongs to, hence:

$$P(\mathbf{A} \mid \mathbf{W}, \mathbf{C}) \;=\; P(\mathbf{A} \mid \mathbf{W}), \tag{4}$$

and this is the criterion that is usually maximized in stochastic based speech recognizers, for instance those using hidden Markov modeling [2] [29] for the acoustic/phonetic decoding. Both the syntactic and the semantic probabilities can be rewritten as:

$$P(\mathbf{W} \mid \mathbf{C}) \;=\; P(w_1) \prod_{i=2}^{N_W} P(w_i \mid w_{i-1}, \ldots, w_1, \mathbf{C}) \tag{5}$$

$$P(\mathbf{C}) \;=\; P(c_1) \prod_{i=2}^{N_W} P(c_i \mid c_{i-1}, \ldots, c_1). \tag{6}$$

We assume that the probability of a word w_i given all the previous words in the sentence and the sequence of concepts $\mathbf{C}$, depends only upon the n most recent words and the concept c_i that the word w_i contributes to express. Under this assumption, equation 5 can be rewritten in terms of the *n-gram concept conditional* word probabilities:

$$P(w_i \mid w_{i-1}, \ldots, w_1, \mathbf{C}) = P(w_i \mid w_{i-1}, \ldots, w_{i-n+1}, c_i) \tag{7}$$

Analogously, the sequence of concept labels $\mathbf{C}$ can be regarded as an $m-th$ order Markov process under the assumption:

$$P(c_i \mid c_{i-1}, \ldots, c_1) \;=\; P(c_i \mid c_{i-1}, \ldots, c_{i-m}) \tag{8}$$

For $n = m = 1$ we can represent the probabilities in equations 7 and 8 as a first order hidden Markov model, whose states represent the conceptual labels and whose observations are the words.

Given the representation of the conceptual structure as a traditional HMM, the decoding of the conceptual content of a sentence can be carried out with the Viterbi algorithm. If the input is a text sentence, Viterbi decoding is used for finding the sequence of states $\tilde{\mathbf{C}}$ such that:

$$P(\mathbf{W}, \tilde{\mathbf{C}}) \;=\; \max_{\mathbf{C}} \mathbf{P}(\mathbf{W}, \mathbf{C}). \tag{9}$$

If the input is speech, we want to find $\tilde{\mathbf{W}}$ and $\tilde{\mathbf{C}}$ that maximize equation 3. In principle, the solution to this problem can be found by substituting each state of the conceptual HMM with a finite state network representing the corresponding bigram structure, and by substituting each word with its corresponding acoustic HMM. The integrated network, obtained in this way, can be used for decoding the input speech with Viterbi algorithm.

3 Implementation of a Speech Understanding System

For putting into practice the idea of MAP conceptual decoding, or at least to show its effectiveness, one needs a corpus of training examples (sentences and their correspondent meaning) and a test set with a criterion for validating the results. A relatively large corpus expressively designed for speech understanding (but not for learning) is being developed within the DARPA ATIS project [13]. ATIS stands for Air Travel Information System and the task is built around a subset of the OAG (Official Airline Guide) database, including 10 American cities. A corpus of spontaneous sentences is being collected and annotated by different sites [21]. The corpus is collected through a Wizard of Oz paradigm. Each subject is given a scenario and a travel planning problem to solve. The subjects are requested to solve the problem by interacting with a machine (that is actually a human *wizard*). The partial and the final responses of the machine are presented to the subjects via a display or a speech synthesizer. The sentences uttered by the subjects are recorded, transcribed and annotated carefully.

Although the ATIS corpus may not be the best corpus for testing a semantic learning paradigm, it is readily available and it includes some kind of meaning annotation that can be indirectly used for our purpose. In this section we will give details about the design and test of a complete speech understanding system for the ATIS task.

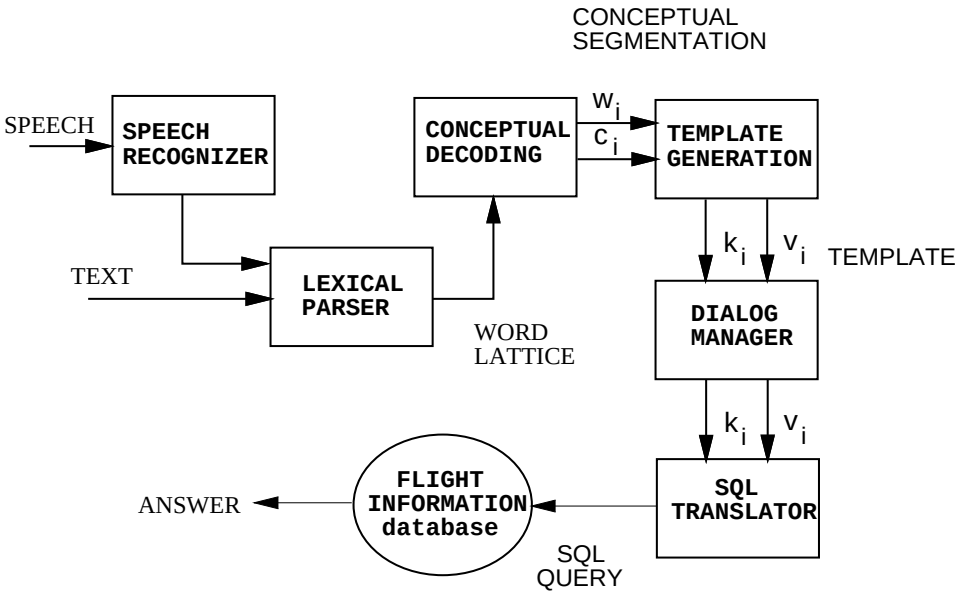

Figure 2: Block diagram of the understanding system

A block diagram of the speech understanding system is shown in Fig. 2.

The system can work both from speech and text input. The *lexical parser* preprocesses sentence transcription in order to assign words to word categories (like numbers and acronyms). The *conceptual decoding* provides a segmentation of the sentence into phrases associated with conceptual units according to the theory explained in chapter 2. The *template generator* assigns to each concept the proper value, while the *dialog manager* takes care of keeping the history of the dialog and including in the current template the missing information. Finally the *SQL* [3] *translator* generates the query. The appropriate information, stored in a relational database, is then retrieved and displayed.

3.1 The Conceptual Decoding

Although MAP decoding of concepts is purely based on semantics, and the final structure of the conceptual model is in principle completely data driven, a certain amount of structure can be imposed during the design of the concept dictionary Γ. Imposing a certain structure to the model certainly alleviates the problems of the locality of the model and that of concept embedding, as explained later. For designing the concept dictionary, we had in mind the typical structure of a query that is represented as in Fig. 3.

In a typical sentence there is a phrase that generally represent the *question* , then a *subject* and finally a *restriction* on the query. For example, in the sentence:

SHOW ME THE FLIGHTS TO SAN FRANCISCO IN THE MORNING

[3]SQL, Structured Query Language, is a high level language for accessing data in a relational database.

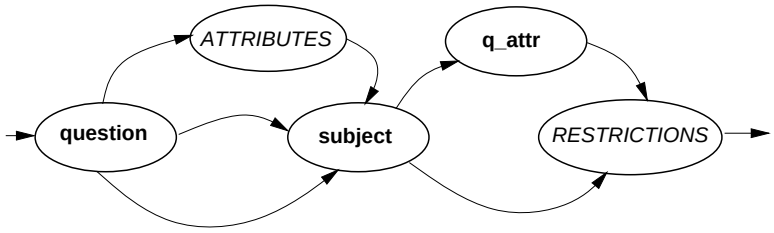

Figure 3: Typical structure of a query

the question phrase is *SHOW ME*, the subject is *THE FLIGHTS* and the restriction is *TO SAN FRANCISCO IN THE MORNING*. The same sentence could be rephrased as:

SHOW ME THE MORNING FLIGHTS TO SAN FRANCISCO

THE MORNING in this sentence plays exactly the same semantic role as the phrase *IN THE MORNING* of the previous sentence, but it has a different syntactic connotation. Therefore when a restriction is placed in front of the subject of the question we call it an *attribute*. Many different concepts can play both the role of restriction and attribute, hence we represent them by separate entities. For instance there is a **fare** concept and an **a_fare** concept that are semantically indistinguishable but with the syntactic role of restriction and attribute respectively. Giving some syntactic connotations to the concepts helps to have better and sharper stochastic models. Without distinguishing between attribute and restriction concepts, the transition probabilities tend to be more uniform and the concept dependent bigram language models tend to be larger since the expressions used for an attribute concept are often different than those used for a restriction concept (*IN THE MORNING* vs. *THE MORNING*).

Another problem is that of concept embedding. The basic assumption we made in section 2, namely that the meaning can be expressed as a sequence of *meaning units* is not generally true [4] Take for instance the following sentence:

WHAT TYPE OF ECONOMY FARE COULD I GET FROM SAN FRANCISCO TO DALLAS

The concept of **question** is represented by two different phrases, namely *WHAT TYPE OF* and *COULD I GET*. Other concepts are embedded in between. Since we are using a flat representation for the meaning (a sequence rather than a tree) one of the possible solution for coping with this problem is to define additional symbols to account for separate phrases in which a single concept can be broken into. For example, in the previous sentence, we introduce a **q_attr** (question attribute) concept, and the corresponding conceptual segmentation will be as follows:

[4]Semantics can be represented, in general, by a tree [1] or by a network [4]. The assumption made here is that the semantics represented by the sentences in the application we are considering is simple enough and can be represented with a flat structure like a sequence of symbols

question:	*WHAT TYPE OF*
a_fare:	*ECONOMY*
subject:	*FARE*
q_attr:	*COULD I GET*
origin:	*FROM SAN FRANCISCO*
destin:	*TO DALLAS*

There are other special concepts like dummy, that accounts for phrases that do not carry information that is relevant for the application (e.g. courtesy forms, etc.), and and that represent conjunctives (e.g. and, or, also, etc.). The concept dictionary used in this implementation counts a total of 48 concepts.

3.2 The Lexical Parser

When dealing with a *speech* understanding system there are different issues related to lexicon that may not be encountered in *text* understanding.

- Although in both systems the vocabulary is generally limited in size, the limitations in a speech system are more severe than in a text system.
- A text system can generally deal with unknown (out of vocabulary) words. Once an unknown word is spotted, the system can take some action, like for instance asking the user to rephrase the sentence using a different word. In a speech recognition system an unknown word cannot generally be detected reliably by the system; it is generally confused and substituted for a known word.
- Numbers and acronyms are generally written in an unambiguous form but they can be uttered in different ways and allow to ambiguous interpretations. For instance, the acronym DC10 can be uttered as *D C TEN* or *D C ONE ZERO*, or *D C ONE OW* and can be interpreted as DC10, or D C10, or DC 10, etc.

The lexical parser analyzes the input transcription and generates a lattice of word categories called *superwords*. A superword can be one of the following:

a. a word, like ABOUT, MONTH, RETURN, etc.

b. a word with optional morphological inflections, like AIRFARE(S), DAY(S), ADVANCE(D), etc. (e.g. there is no distinction between AIRFARE and AIRFARES; both words are represented by the super-word AIRFARE(S)).

c. a grammar, represented by a finite state automaton (FSA); for instance, the grammar for natural numbers (e.g. THIRTY SEVEN), the grammar for airport acronyms (e.g. D F W), the grammar for airport names (e.g. SAN FRANCISCO INTERNATIONAL AIRPORT), etc. Each sequence recognized by an FSA is characterized by the corresponding FSA identifier and a normalized form of the compound word (e.g. THIRTY SEVEN is represented as ((number)37), D F W as ((airport)DFW)). As far as the stochastic model is concerned, two words with the same FSA identifier are represented by the same super-word.

Experiment Description	% of correct concepts	% of correct sentences
no super-lexicon	94.1	80.4
super-lexicon	96.8	88.5

Table 2: Effect of the super-lexicon in conceptual decoding

When there is ambiguity in deciding which superword to assign to a sequence of words, the lexical parser generates all the possible interpretations and arranges them in a lattice [5].

Besides, articles (i.e. *THE* and *A*) are deleted by the lexical parser, hence they play no role in the conceptual decoding process. Concept conditional language models are thus estimated among content words only. The reason for doing this is that in this particular application there is no relevant information carried by an article in front of a noun or an adjective. Besides, articles (like other short function words) can be easily misrecognized or deleted by the speech recognizer. Hence, due to the locality of the language model (i.e. bigrams of words), many bigrams will carry the probability of a noun or an adjective preceded by an article, missing the more important correlation to the preceding content word.

Using super-words reduces the number of parameters to be estimated and increases the robustness of the system. The probabilities associated to a given super-word are shared between all the words that are represented by the same super-word. This has the effect of allowing the system to generalize the statistics gathered in the training phase to all the words belonging to the same super-word. This is shown in an experiment where we implemented the conceptual decoding in two different situations. In the first implementation there was no super lexicon, the vocabulary consisted of 501 words, including three word classes (i.e. numbers, city names, and acronyms), and the articles were accounted for in the bigrams. In this case, the input to the system was a textual transcription of each utterance, where the numbers and the acronyms were unambiguously represented (e.g. *What's the fare for US-AIR 4393*). In the second experiment a super-lexicon including 753 words and 18 grammars was used. In this case the input to the system was a speech transcription of each utterance. It has to be noticed that although the lexicon in the second experiment was larger than the lexicon in the first experiment, the word coverage over the test set was the same in both cases. The test set consisted of 148 context independent sentences (included in the official February 1991 test set [16]). The 148 sentences were hand segmented into concepts. The set of sentences corresponded to an overall number of 713 concepts. The performance were assessed based on the hand segmentation. The percentage of correct concepts (when the conceptual segmentation agreed with the one provided by hand) and of correct sentences (sentence for which all the included concepts are correctly decoded and segmented) is reported in Table 2.

[5] It is quite straightforward to modify the Viterbi algorithm for performing the decoding from a lattice rather than from a sequence of observations. A discussions on this subject can be found in [7]

3.3 Interfacing with a Speech Recognizer

The most natural way of interfacing the conceptual decoder with a speech recognizer is by implementing the maximization of equation 3. This requires to implement a decoder that explores a network obtained by explicitly instantiating acoustic HMMs [11] representing words (or superwords) of the vocabulary for any concept. For a task like ATIS the dimension of the resulting network can be rather large. In theory, if there are 50 conceptual states and about 1,000 words, each one of them represented by a HMM with an average number of 15 states, the overall network is bound by a total number of 750,000 acoustic HMM states, with a number of connections of the order of 50,000,000 (each conceptual conditional bigram model is represented by 1000×1000 connections). Of course not all the bigrams are observed or even possible in each state. If only those words and bigrams that were observed during the training are represented in a conceptual state, a more reasonable model can be obtained. In an experimental version of CHRONUS we estimated an integrated model with a total of 2400 HMM word models (corresponding to about 36,000 HMM acoustic states) and nearly 46,000 connections. This size of the model can be easily managed by a beam search recognizer [12]. The problem in using such a model is that while it constitute a reasonably good model for decoding the semantic message of a sentence, the limited amount of training data used for its estimation makes it a quite coarse model for constraining the speech recognition process. When bigrams of words that were not observed in the training data are actually uttered, the recognizer is forced to substitute them for known bigrams. Hence the recognition errors are propagated along the sentence, resulting in relatively poor recognition performance.

Smoothing techniques can be applied for estimating the probability of unobserved bigrams, like for instance methods relying on the Good-Turing estimation of probabilities [6]. This will increase the complexity of the model by allowing all the possible bigrams in each state. However a factorization of the maximization of equation 3 [18] can still lead to reasonably good results at an acceptable complexity. Hence several solutions could be implemented, like best first coupling (the best first recognized sentence is given to the conceptual decoding), N-best coupling [15] and word lattice coupling [9].

3.4 The Template Generator

The goal of the template generator module is that of analyzing the conceptual segmentation and generating the final representation of the meaning (i.e. the *template*) by supplying the correct value to each concepts detected during the decoding stage. For every relevant concept a look-up table was built for performing the mapping between phrase templates and conceptual values. The look-up table associates keywords or short phrases to concept values. Values are then assigned to the decoded concepts according to the result of a pattern matching procedure with the keyword stored in the appropriate tables. The values that are eventually associated to the decoded concepts belong to different categories. They can be actual database items (e.g. **Boston, American airlines, breakfast**), database attributes, (e.g. **flight, stop, meal**), logic values (e.g. **null, not null**) or operators, (e.g. **minimum, maximum**). The design of pattern matching tables is still manual. More details on the template generator can be found in [26].

3.5 The Dialog Manager

The *dialog manager* implements the function of keeping the dialog history and allowing the resolution of anaphoric and elliptical sentences. The simple strategy implemented in our system consists in keeping a current *context template* with all the information that has been used for specifying the actual query. When a new sentence is presented to the system, the dialog manager tries to merge the current template with the current context template, in order to get the missing information. The merging of the two templates follows *application specific* rules. For instance, when a concept is mentioned in a new sentence, with a different value than the corresponding concept in the context template, all concepts in the context template at a lower hierarchical level are deleted. This assumes a predefined hierarchy of the concepts. The origin and the destination of a flight have the highest level in the hierarchy. Then, when either the origin or the destination are different from those specified in the context template, the context template is deleted and a new context is started. This strategy, although very simple, has proven to be effective in the majority of sentences of this task.

3.6 The SQL translator

The last part of the interpretation process, namely the access to the required information, is implemented through a translator that dynamically generates the SQL query in order to retrieve the data. The template produced by the template generator is processed according to the value of its subject concept. If a subject concept is not found in the template a default subject is used. The subject of the query is used for selecting the right table of the database. If there is more than one subject or the subject is not directly related to a particular table, a link function is invoked in order to perform the correct joins. Once the table (or a joint table) is selected, the rest of the template tokens are interpreted accordingly.

4 Putting it into Practice

Assessing the performance of a language understanding system is still an open problem mainly because the concept of *correct answer* is generally ambiguous and must be based on defined conventions that are not task independent. The DARPA community agreed upon scoring answers by comparison with given reference answers that are produced for each valid sentence of the corpus. The answers can be made up of data extracted from the flight information database, numbers, or logical values (yes/no). In case of ambiguity of the question, multiple reference answers are given. Of course the problem of the definition of a *correct* answer still remains. For instance, for a question like

SHOW THE LATE EVENING FLIGHTS BETWEEN BOSTON AND DALLAS

the correctness of the answer depends upon the conventional definition of *late evening*. Then, once a time interval has been defined for late evening, it is still not clear what is the information to be listed. It could be the airline and flight number of each flight, but it could also include the departure time, the arrival time, the fare, and so on. A special committee within the DARPA

Error type	Number of Sentences
Conceptual decoding	30
Template generation	19
Dialog manager	20
SQL translator	52

Table 3: Analysis of the errors for the NL ATIS February 92 test

community agreed upon a certain number of rules, called *principles of interpretation* [21], that should rule the majority of cases. Besides, it was also agreed on using two reference answers, namely a *minimal* and a *maximal* reference answer. An answer is thus considered correct if it contains all the information included in the minimal reference answer and no more than the information included in the maximal reference answer.

4.1 Experimental Performance

The described system was tested on a set of 687 sentences called February 92 test set [22]. For increasing the robustness we provided the system with a simple rejection criterion. The rejection heuristic is based on the measure of success in the operation of template generation (how many decoded concepts are successfully matched to a value), although more sophisticated heuristics can be developed. The results [24] on the complete test set, from text input, account for 68% of correct answers, 18% wrong answers and 14% rejects. When the system was coupled with a speech recognizer through the best first hypothesis, the performance dropped to 52% of correct answers, 26% wrong answers and 22% rejects. However, the contribution of the different modules to the overall error rate is far more interesting than it absolute value. All the 121 sentences that produced a wrong answer (in the text understanding experiment) were carefully analyzed and the errors were classified according to Table 3. From this analysis it results that most of the errors are due to the parts of the system that are not trained. The template generator errors reflect a lack of entries in the look-up tables. The dialog manager errors are due to the fact the the simple strategy for merging the context and the current template should be refined with more sophisticated rules. The SQL translator should be an error-free module. Its only function is that of translating between two different representation in a deterministic fashion. However, in this test, the SQL module faced two kinds of problems. The first is that the interpretation rules used for generating the answers were not exactly the same ones used for the official test, and this accounts for roughly half of the errors. The other half of the error is due to the limited power of the template representation, and this will be discussed in section 5.

4.2 Training the Conceptual Model

4.2.1 Smoothing of Bigram Models

The conceptual model, as explained in section 2, is defined by two sets of probabilities, namely the *concept conditional bigrams* $P(w_i \mid w_{i-1}, c_i)$ and the *concept transition probabilities* $P(c_i \mid c_{i-1})$. In the first experiments these probabilities were estimated using a set of 532 sentences whose conceptual segmentation was provided by hand. The accuracy of the system in the experiments carried out using the model estimated with such a small training set, although surprisingly high [17], shows a definite lack of training data. Smoothing the estimated model probability provides an increase of the performance. The knowledge of the task can be introduced through a *supervised smoothing* of the concept conditional bigrams. The supervised smoothing is based on the observation that, given a concept, there are several words that carry the same meaning. 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.

A number of groups of synonyms were manually compiled for each concept. The occurrence frequencies inside a group were equally shared among the constituting words, giving the same bigram probability for synonymous words.

4.2.2 Using a Larger Training Corpus

If one wants to use a larger corpus than the initial handlabeled few hundred sentences and wants to avoid an intensive hand segmentation labor, one has to capitalize on all the possible information associated to the sentences in the corpus. Unfortunately, when the corpus is not expressively designed for learning, like the ATIS corpus, the information needed may not be readily available. In the remaining of this section we analyze solutions that, although particularly devised for ATIS, could be generalized to other corpora and constitute a guideline for the design of new corpora.

A training token consists of a sentence and its associated meaning. The meaning of sentences in the ATIS corpus is not available in a declarative form. Instead, each sentence is associated with the *action* resulting from the *interpretation* of the meaning, namely the correct answer. One way of using this information for avoiding the handlabeling and segmentation of all the sentences in the corpus consists in creating a training loop in which the provided correct answer serves the purpose of a feedback signal. In the training loop all the available sentences are analyzed by the understanding system obtained with an initial estimate of the conceptual model parameters. The answers are then compared to the reference answers and the sentences are divided into two classes. The *correct* sentences, for which we assume that the conceptual segmentation obtained with the current model is correct, and the *problem sentences*. Then the segmentation of the correct sentences is used for reestimating the model parameters,

and the procedure is repeated again. The procedure can be repeated until it converges to a stable number of correct answers. Eventually, the remaining *problem sentences* are corrected by hand and included in the set of correct sentences for a final iteration of the training algorithm. This procedure proved effective for reducing the amount of handlabeling. In the experiment described in [25] we showed that the performance increase obtained with the described training loop, without any kind of supervision (the remaining *problem* sentences were excluded from the training corpus) is equivalent to that obtained with the supervised smoothing. This means that the training loop, although is not able to learn radically new expressions or new concepts, is able to reinforce the acquired knowledge and to *infer* the meaning of semantically equivalent words. In a set of 4500 sentences the training loop automatically classified almost 80% of the sentences, leaving the remaining 20% to the manual segmentation.

4.2.3 The Sequential Correspondence Assumption

In section 2 we based our formalization of the speech understanding problem on the assumption that there is a sequential correspondence between the representation of a sentence (words or acoustic measurements) and the corresponding representation of meaning. This assumption is not generally true for any translation (semantic or not) task. An interesting example (reported in [27]) of a task where there is no sequential correspondence between a message and its semantic representation, is that of roman numbers (e.g. I, XXIV, XCIX) and their correspondent decimal representation (e.g. 1, 24, 99). Fortunately, in a natural language understanding task, we may have the freedom of choosing the semantic representation, like we did in the implementation of CHRONUS explained above. But in general, if we are dealing with a large corpus of sentences that have not been expressively designed for the purpose of learning a semantic translator, and we would like to take advantage of some kind of semantic annotation already available, we may have to face the problem of the not sequentiality of the representation. For instance, in the ATIS corpus, each sentences is associated with the intermediate representations used by the annotators for obtaining the reference correct answers. In fact the annotators rephrase each valid sentence in an artificial language that is a very restricted form of English. This *pseudo-English* rephrasing (called *win* or *wizard input*) constitute the input of a parser, called NLparse [10], that unambiguously generates the SQL query. For instance, for a sentence like:

> *I'D LIKE TO FIND THE CHEAPEST FLIGHT FROM WASHINGTON D C TO ATLANTA*

The *win* rephrasing is:

> *List cheapest one direction flights from Washington and to Atlanta*

and the corresponding associated SQL statement is:

> (SELECT DISTINCT flight.flight_id FROM flight WHERE (flight.flight_id IN (SELECT flight_fare.flight_id FROM flight_fare WHERE flight_fare.fare_id IN (SELECT fare.fare_id FROM fare WHERE fare.one_direction_cost = (SELECT MIN (fare . one_direction_cost) FROM fare WHERE fare.fare_id IN (SELECT flight_fare.fare_id FROM flight_fare WHERE flight_fare.flight_id IN (SELECT flight.flight_id FROM flight WHERE (flight.from_airport IN (SELECT airport_service.airport_code FROM airport_service WHERE airport_service.city_code IN (SELECT city.city_code FROM city WHERE city.city_name = 'WASHINGTON')) AND flight.to_airport IN (SELECT airport_service.airport_code FROM airport_service WHERE airport_service.city_code IN (SELECT city.city_code FROM city WHERE city.city_name = 'ATLANTA')))))))) AND (flight.from_airport IN (SELECT airport_service.airport_code FROM airport_service WHERE airport_service.city_code IN (SELECT city.city_code FROM

```
city WHERE city.city_name = 'WASHINGTON' )) AND flight.to_airport IN (SELECT airport_service.airport_code FROM
airport_service WHERE airport_service.city_code IN (SELECT city.city_code FROM city WHERE city.city_name = 'AT-
LANTA' )))));.
```

Both the SQL query and the *win* sentence can be considered semantic representations of the original sentence. In fact the SQL query is the final target of the understanding system and can be unequivocally obtained from the *win* sentence through an existing parser. Obviously the sequential correspondence assumption is strongly violated for the SQL representation. However a sequential correspondence can be easily found between the pseudo-English *win* sentence and the original message, at least for the shown examples. Since all the valid sentences in the ATIS corpus have a *win* annotation, the pseudo-English language can be thought of as an alternate candidate for the meaning representation in our learning framework. Using *win* for representing the meaning may lead to two different solutions. In the first we can think of developing a system that learns how to translate natural language sentences into pseudo-English sentences and then use the existing parser for generating the SQL query. In the second solution each *win* sentence in the corpus can be translated in the corresponding conceptual representation used for CHRONUS. This translation is unambiguous (*win* is an unambiguous artificial language by definition). A parser can be easily designed for performing the translation or, simply use CHRONUS itself for performing the translation[6]. Unfortunately also for the *win* representation, the sequential correspondence assumption is violated for a good percentage of the sentences in the corpus. A typical example is constituted by the following sentence:

> *COULD YOU PLEASE GIVE ME INFORMATION CONCERNING AMERICAN AIRLINES A FLIGHT FROM WASHINGTON D C TO PHILADELPHIA THE EARLIEST ONE IN THE MORNING AS POSSIBLE*

whose corresponding *win* annotation is:

> *List earliest morning flights from Washington and to Philadelphia and American.*

The problem of reordering the words of *win* representation for aligning it with the original sentence is a complex problem that cannot be solved optimally. Suboptimal solutions with satisfactory perfomance can be developed based on effective heuristics. We will not discuss the details of how the reordering can be put into practice. Rather we want to emphasize the fact that an iterative algorithm based on a model similar to that explained in section 2 led to almost 91% correct alignments between English sentences and corresponding *win* representations on a corpus of 2863 sentences. With additional refinements this technique can be used, integrated in the training loop, for automatically processing the training corpus of the conceptual model.

5 Discussion and Conclusions

In this paper we propose a new paradigm for language understanding based on a stochastic representation of semantic entities called concepts. An interesting way of looking at the language understanding paradigm is in term of a language translation system. The first block in

[6] *win* is a subset of natural English. Only a little adaptation was needed for developing a *win* translator based on the existing CHRONUS

Fig. 1 translates a sentence in natural language (*N-L*) into a sentence expressed in a particular semantic language (*S-L*). The natural language characteristics are generally unknown, while the semantic language designed to cover the semantic of the application is completely known and described by a formal grammar. The second step consists in the translation of the sentence in *S-L* into computer language code *C-L* for performing the requested action. This second module can be generally (but not necessarily) designed to cover all the possible sentences in *S-L*, since both *S-L* and *C-L* are known. However, the boundary between the first and the second module is quite arbitrary. In [28], for instance, an automatic system is designed for translating ATIS English sentences directly into the SQL query, and in [14] there is an example of a system that goes from an English sentence to the requested action without any intermediate representation of the meaning. However, the closer we move the definition of *S-L* to *N-L*, the more complicate becomes the design of the *action transducer*, reaching in the limit the complexity of a complete understanding system. Conversely, when we move the definition of *S-L* closer to *C-L*, we may find that learning the parameters of the *semantic translator* becomes quite a difficult problem when the application entails a rather complex semantics. The subject of this paper deal with the investigation of the possibility of automatizing the design of the first block (i.e. the *semantic translator*) starting from a set of examples. The semantic language chosen for the experiments reported in this paper is very simple and consists of sequences of keyword/value pairs (or *tokens*). There is no syntactic structure in the semantic language we use. Two sentences for which the difference in the semantic representation is only in the order of the tokens are considered equivalent. In this way we cover a good percentage of sentences in the domain, but still there are sentences that would require a structured semantic language. For instance the two following sentences are indistinguishable when represented by our semantic language, and obviously they have a different meaning.

IS THE EARLIEST FLIGHT GOING TO BOSTON ON A SEVEN FOUR SEVEN

IS THE EARLIEST FLIGHT ON A SEVEN FOUR SEVEN GOING TO BOSTON

The representation of this kind of sentences requires a more sophisticated semantic language that allows the use of bracketing for delimiting the scope of modifiers.

Although the system we propose uses a very simple intermediate semantic representation, we showed that it can successfully handle most of the sentences in a database query application like the ATIS task. When this simple representation is used and when the problem of *semantic translation* is formalized as a communication problem, a MAP criterion can be established for decoding the *units of meaning* from text or speech. The resulting decoder can then be integrated with other modules for building a speech/text understanding system.

An understanding system based on a learning paradigm, like the one proposed in this paper, can evolve according to different dimensions of the problem. One dimension goes with the increase in complexity of the semantic language *S-L*. Rather than using a sequential representation on could think of a tree representation of the meaning. However, this poses additional problems both in the training and decoding stage, and requires the use of algorithms designed for context-free grammars, like for instance the *inside-outside* algorithm [3][23] that have a higher complexity that those explained in this paper. Another dimension of the problem goes

toward a complete automatization of the system, also for those modules that, at the moment, require a manual compilation of some of the knowledge sources. One of these modules is the *template generator*. Both [20] and [28] report examples of systems where the decision about the actual values of the conceptual entities (or an equivalent information) is drawn on the basis of knowledge acquired automatically from the examples in the training corpus. The kind of annotation required for the training corpus is also another dimension along with the research on learning to understand language should move. A strategy for learning the understanding function of a natural language system becomes really effective and competitive to the current non-learning methods when the amount of labor required for annotating the sentences in a training corpus is comparable or inferior to the amount of work required for writing a grammar in a traditional system. This requires the development of a learning system the does not require any other information than the representation of the meaning associated to each sentence (e.g. it does not require an initial segmentation into conceptual units, like in CHRONUS, for bootstrapping the conceptual models). Moreover, the representation of the meaning should be made using a *pseudo-natural* language, for making easier and less time consuming the work of the annotators. An example of this kind of annotation was introduced in section 4.2.3 with the pseudo-English *win* rephrasing. This suggests a possible evolution of the learning strategy for understanding systems toward a system starting with the limited amount of knowledge required for understanding a small subset of the whole language (e.g. the *win* language). Then the system can evolve to understanding larger subsets of the language using the language already acquired for rephrasing new and more complex examples. But, of course, the science of learning to understand is still in its infancy, and many more basic problems must be solved before it becomes an established solution to the design of a language interface.

References

[1] Minski, M., "A Framework for Representing Knowledge," in P. Winston (ed.). *The Psychology of Computer Vision,* New York: McGraw-Hill, 211-277, 1975.

[2] Jelinek, F., "Continuous Speech Recognition by Statistical Methods," *Proceedings of IEEE,* vol. 64, no. 4, pp. 532-556, 1976.

[3] Baker, J., "Trainable Grammars for Speech Recognition," in Wolf, J. J., Klatt, D. H., editors, *Speech communication papers presented at the 97th Meeting of the Acoustical Society of America,* MIT, Cambridge, MA, June 1979.

[4] Sowa, J., *Conceptual Structures.* Reading, MA:Addison-Wesley, 1984.

[5] Allen, J, "Natural Language Understanding," The Benjamin/Cummings Publishing Company, Inc., 1987.

[6] Katz, S. M., "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *IEEE Trans. on Acoustic, Speech and Signal Processing,* Vol. ASSP-35, No. 3, March 1987.

[7] Fissore, L., Laface, P., Micca, G., Pieraccini, R., "Lexical Access to Large Vocabularies for Speech Recognition," *IEEE Trans. on Acoustics, Speech and Signal Processing,* Vol. 37, No. 8, August 1989.

[8] Brown, P. F, Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J., Mercer, R. L., Roosin, P. S., " A Statistical Approach to Machine Translation," *Computational Linguistics*, Volume 16, Number 2, June 1990.

[9] Fissore, L., Kaltenmeier, A., Laface, P., Micca, G., Pieraccini, R., "The Recognition Algorithms," in *Advanced Algorithms and Architectures for Speech Understanding,* G. Pirani editor, Springer-Verlag Brussels -Luxemboug, 1990.

[10] Hemphill, C. T., Godfrey, J. J., Doddington, G. R., "The ATIS Spoken Language Systems, pilot Corpus," *Proc. of 3rd DARPA Workshop on Speech and Natural Language*, pp. 102-108, Hidden Valley (PA), June 1990.

[11] 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.

[12] Pieraccini, R., Lee, C. H., Giachin, E., Rabiner, L. R., "An Efficient Structure for Continuous Speech Recognition," in *Speech Recognition and Understanding, Recent Advances, Trends and Applications,* P. Laface and R. De Mori editors, Springer-Verlag Berlin Heidelberg, 1992.

[13] 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.

[14] Gorin, A. L., Levinson, S. E., Gertner, A., "Adaptive Acquisition of Language," *Computer Speech and Language*, **5**, pp. 101-132, 1991.

[15] 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*, Pacific Grove, (CA), February 1991.

[16] Pallett, D. S., "Session 2: Darpa Resource Management and ATIS Benchmark Test Poster Session," *Proc. of 4th DARPA Workshop on Speech and Natural Language*, Pacific Grove, (CA), February 1991.

[17] 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*, Pacific Grove, (CA), February 1991.

[18] 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.

[19] Prieto, N., Vidal, E., "Learning Language Models through the ECGI Method," *Proc. of EUROSPEECH 91,* Genova, Italy, September 1991.

[20] Giachin, E. P, "Automatic Training of Stochastic Finite-State Language Models for Speech Understanding", *Proc. of International Conference on Acoustics, Speech and Signal Processing*, ICASSP-92, San Francisco, CA, March 1992.

[21] *MADCOW*, "Multi-Site Data Collection for a Spoken Language Corpus,", *Proc. of Fifth Darpa Workshop on Speech and Natural Language*, Harriman, NY, Feb 1992.

[22] Pallett, D. S., Dahlgren, N. L., Fiscus, J. G., Fisher, W. M., Garofolo, J. S., Tjaden, B. C., "DARPA February 1992 ATIS Benchmark Test Results," *Proc. of Fifth Darpa Workshop on Speech and Natural Language*, Harriman, NY, Feb 1992.

[23] Pereira, F., Schabes, Y., "Inside-Outside Reestimation from Partially Bracketed Corpora," *Proc. of Fifth Darpa Workshop on Speech and Natural Language*, Harriman, NY, Feb 1992.

[24] Pieraccini, R., Tzoukermann, E., Gorelov, Z., Levin, E., Lee, C.-H, Gauvain, J.-L, "Progress Report on the Chronus System: ATIS Benchmark Results," *Proc. of Fifth Darpa Workshop on Speech and Natural Language*, Harriman, NY, Feb 1992.

[25] Pieraccini, R., Gorelov, Z., Levin. E., Tzoukermann, E., "Automatic Learning in Spoken Language Understanding," *Proc. of 1992 International Conference on Spoken Language Signal Processing*, ICSLP 92, Banff, Alberta, Canada, October 1992.

[26] Tzoukermann, E., Pieraccini, R., Gorelov, Z., "Natural Language Processing in the CHRONUS System," *Proc. of 1992 International Conference on Spoken Language Signal Processing*, ICSLP 92, Banff, Alberta, Canada, October 1992.

[27] Oncina, J., García, P., Vidal, E., "Learning Subsequential Transducers for Pattern Recognition Interpretation Tasks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 5, pp. 448-458, May 1993.

[28] Kuhn, R., De Mori, R., "Learning Speech Semantics with Keyword Classification Trees," *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing,* ICASSP-93, Minneapolis, Minnesota, April 27-30, 1993,

[29] Rabiner, L. R., Juang, B. H., "Fundamentals of Speech Recognition," Prentice Hall Signal Processing Series, 1993.