IMPROVING SPEECH UNDERSTANDING BY INCORPORATING DATABASE CONSTRAINTS AND DIALOGUE HISTORY

Frank Seide, Bernhard Rueber, Andreas Kellner

Philips GmbH Research Laboratories Aachen, Weisshausstr. 2, D-52066 Aachen, Germany
E-mail: {seide,rueber,kellner}@pfa.research.philips.com

ABSTRACT

In the course of a (man-machine) dialogue, the system's belief concerning the user's intention is continuously being built up. Moreover, restricting the discourse to a narrow application domain further constrains the variety of possible user reactions. In this paper, we will show how these knowledge sources may be utilized in a stochastic framework to improve speech understanding. On field-test data collected with our automatic exchange board prototype PADIS[1], a relative reduction of attribute errors by 27% has been obtained.

1. INTRODUCTION

In an automatic inquiry system, the computer conducts a dialogue to find out the user's *dialogue goal*. It then carries out the desired action, which is typically to perform some transaction on a certain entry of a database. For example, in an automatic exchange board system, the user's dialogue goal could be to be connected to some person specified by name. The system would have to find out which person, and that call completion is desired. It would then retrieve the phone number from the database and transfer the call.

In state-of-the-art systems, domain knowledge is incorporated into the speech recognizer by a word-level language model, which is typically a hybrid of a phrase and/or word N-gram and a stochastic context-free grammar [1] or a finite-state network [2, 3, 4]. Common to systems of this type is that the word-level model actually models not only the a-priori probabilities of word sequences, but also, implicitly, the distribution of the dialogue goals.

When designing our automatic exchange board system, we found some useful long-span constraints. For instance, we can exploit that a cooperative user will never intentionally ask for combinations of first and last name and/or affiliation that do not refer to an *existing* database entry. However, these constraints cannot be captured by the word-level models described above, but must be modelled separately (they may even span across multiple dialogue turns).

The PADIS prototype is based on the Philips automatic inquiry system [5], which understands natural-language requests in fluently spoken continuous speech over the telephone and conducts a mixed-initiative dialogue. We have extended this system to directly incorporate

- an explicit a-priori distribution of the user's dialogue goal (as far as separable from the word-level model),
- dependencies on the prompt presented to the user, and
- dependencies on the items already stated (or at least the system's belief on that)

into our maximum-a-posteriori (MAP) criterion. This leads to a significant improvement in understanding accuracy and has the practical advantage that the database can be exchanged or maintained without language-model retraining.

2. PROBABILISTIC FRAMEWORK

Our system is based on a speech production model as depicted in figure 1. The user has a certain dialogue goal G in talking to the system (which we regard constant during a dialogue since we assume a cooperative user).

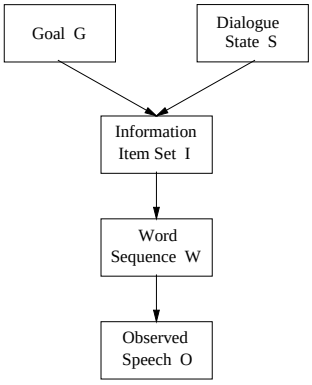

Figure 1: Production model of a user's utterance.

In every utterance, the user states a set of information items I, which is determined by G and the current dialogue state S. S includes the system's current question and the system's current belief on what has already been stated by the user. It can be directly observed, while G is a hidden variable. I is then cast into a sequence of words W and finally observed

[1] Philips Automatic Directory Information System

by our system as a sequence of acoustic feature vectors O. The symbols are summarized in table 1.

Note that the user's goal G and the dialogue state S are indeed not independent of each other but coupled via the information collected in previous turns.

Table 1: Summary of symbols with examples.

Sym.	Explanation	Example
G	dialogue goal	{ talk to S. White }
G_{LM}	LM-modelled part	{ talk to }
G_{DB}	DB-modelled part	{ S. White }
S	system status	current belief = { Sally } question = *Which Sally ?*
W	word sequence	`give me Doctor White please`
W^T	word seq. templ.	`give me <title> <name> please`
I	inform. item set	{ `@req=connect`, `@title=Doctor`, `@name=White` }
I^T	inform. template	{ `@req=connect`, `@title=<title>`, `@name=<name>` }
O	acoustic obs.	(observed feature vectors)

2.1. MAP Criterion for Speech Understanding

We define the speech understanding task as finding the information item set $\hat{I}$ which most probably generated our acoustic observation O, given the system's current state S (maximum-a-posteriori criterion):

$$\begin{aligned} \hat{I} &= \arg\max_I P(I|SO) \qquad (1)\\ &= \arg\max_I \sum_{W,G} p(OWISG) \\ &= \arg\max_I \sum_W p(O|W) \cdot \sum_G P(WISG) \end{aligned}$$

where W is the underlying word sequence and G the (unknown) dialogue goal. We make the model assumption that $P(WISG)$ is 0 for all I inconsistent with S or itself, but otherwise independent of S and thus proportional to $P(WIG)$:

$$\begin{aligned} P(WISG) &= P(WIG) \cdot \delta_{I,S}\ \delta_{I,I} \cdot \alpha \qquad (2)\\ \delta_{X,Y} &= \begin{cases} 1 & \text{if } X \text{ is consistent with } Y \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

(α is a normalization constant which is independent of I, and can be ignored since it does not contribute to the maximization). As a simplification, we replace the sum over W by the maximum[2] and obtain:

$$\hat{I} = \arg\max_I \left\{ \max_W p(O|W) \cdot \sum_G P(WIG) \cdot \delta_{I,S}\ \delta_{I,I} \right\}$$

2.2. Modelling the Goal's Distribution

We decompose the goal G into two parts (subsets) which have to be evaluated differently: the *language-model goal* G_{LM} and the *database goal* G_{DB}. They are disjoint, and their priors are assumed to be independent, i.e.

$$P(G) = P(G_{LM}) \cdot P(G_{DB}) \qquad (3)$$

$$\begin{aligned} \sum_G P(WIG) &= \sum_G P(WI|G)\, P(G) \\ &= \sum_G P(WI|G_{LM}G_{DB})\, P(G_{LM})\, P(G_{DB}) \end{aligned}$$

The language-model goal G_{LM} is the part of the goal for which the prior $P(G_{LM})$ is not explicitly available but implicitly modelled by the language model. If, for example, the goal G is to talk to a certain person, G_{LM} would be { get connected to some subscriber }. We get:

$$\begin{aligned} \sum_G P(WIG) &= \sum_G P(WIG_{LM}|G_{DB}) \cdot P(G_{DB}) \\ &= \sum_{G_{DB}} P(WI|G_{DB}) \cdot P(G_{DB}) \end{aligned}$$

On the other hand, for the database goal G_{DB}, the a-priori distribution $P(G_{DB})$ is explicitly available. $P(G_{DB})$ reflects how often a person is asked for, and is 0 for non-existing database entries.

The language model, however, still models the sentence *structure*. For example, it does *not* model *whom* to talk to, but *how* the person is specified: by first and last name, by title and last name, by first name only, etc.

Thus, the language model provides a-priori probabilities only for word-sequence and information-item *templates*, in which all G_{DB}-related words are replaced by a placeholder (e.g. "`I want to talk to <fname> <name>`"). W^T and I^T denote the templates for W and I, respectively. Thus,

$$\begin{aligned} \sum_G P(WIG) &= \sum_{G_{DB}} P(WIW^TI^T|G_{DB}) \cdot P(G_{DB}) \\ &= P(W^TI^T) \cdot \sum_{G_{DB}} \delta_{WI,G_{DB}} \cdot P(G_{DB}), \end{aligned}$$

because for every (W^T, I^T, G_{DB}) there is exactly one (W, I), and $P(W^TI^T)$ can be assumed independent of G_{DB} (since all G_{DB} are of the same type). $P(W^TI^T)$ is estimated and evaluated by a stochastic grammar without further splitting into W^T and I^T, like in [1].

2.3. Decision Rule

We obtain the final decision rule:

$$\begin{aligned} \hat{I} \approx \arg\max_I \Bigg\{ \max_W \underbrace{p(O|W)}_{\text{acoustics}} \cdot \underbrace{P(W^TI^T)}_{\text{grammar}} \qquad (4)\\ \cdot \underbrace{\sum_{G_{DB}} \delta_{WI,G_{DB}} \cdot P(G_{DB})}_{\text{database knowledge}} \cdot \underbrace{\delta_{I,S}\ \delta_{I,I}}_{\text{consistency constraints}} \Bigg\} \end{aligned}$$

[2] The maximization over W can actually be restricted to word sequences whose interpretation is I, because for all other W, $P(WIG)$ is zero.

3. IMPLEMENTATION

To allow for an economic usage of CPU resources, our system uses a multistage approach to implement the decision rule. From stage to stage, the number of alternative hypotheses to consider becomes less, while the model becomes more and more complex.

3.1. Speech Recognition

The first stage is a speech recognizer as described in [6]: The *word-hypotheses generator* uses the acoustic model $p(O|W)$ and a word-unigram language model to identify and score plausible word hyptheses. The *word-graph optimizer* then combines them, using a bigram, into a pruned word graph, which is the output of this stage. A word graph is a compact representation of plausible alternative sentence hypotheses. Every path through the graph is a sentence hypothesis.

3.2. Natural Language Processing

In this stage, the word graph is parsed with an attributed stochastic grammar and converted into an information graph (see [1]). It represents all possible interpretations of the paths through the word graph. The grammar provides the language-model probability $P(W^T I^T)$ for every path through the information graph, implicitly incorporating $P(G_{LM})$.

3.3. N-Best Rescoring

In the third stage, the decision on the most likely $\hat{I}$ is taken, considering the database goal's a-priori distribution $P(G_{DB})$ and the consistency constraints $\delta_{I,S}$ and $\delta_{I,I}$. We employ the N-best algorithm described in [7]. Its special feature is that N does not have to be known in advance, but the N best sentences can be computed one after the other, sorted by their scores delivered by the third stage.

Every sentence hypothesis gets rescored using the full model, where inconsistent paths or paths not referring to valid database entries are immediately rejected. The N-best search stops if the most likely hypothesis is expected to have been considered. To retain real-time operation, it also stops if no consistent path can be found amongst the top $N_{max} = 100$ hypotheses. In this case, the result is the empty set $\hat{I} = \emptyset$.

Consistency with the Database

Consistency with the database is ensured by the a-priori probability $P(G_{DB})$: For an information item set I of a sentence hypothesis that does not refer to at least one existing database entry, $P(G_{DB})$ will be 0, and therefore the hypothesis is rejected.

The following example[3]illustrates this (Marvin Jones is a valid database entry, whereas Martin Jones is not). The notation `Martin:fname` means that the attribute `fname` (first name) was assigned the value `Martin` by the grammar.

Best path:
`I want to talk to Martin:fname Jones:name please`
db input: last name: Jones
first name: Martin
⇒ No database match: hypothesis gets rejected.

2nd best path:
`I want to talk to Marvin:fname Jones:name please`
db input: last name: Jones
first name: Marvin
⇒ Consistent.
This path is selected.

Matching Rules

The consistency constraints $\delta_{I,S}$ and $\delta_{I,I}$ are implemented by *matching rules*. In the following, we explain the rules actually employed in our prototype.

Rule 1: Consistency Within Interpretation Itself

A hypothesized interpretation of the user's response is only accepted if it does not contain contradictory values for an attribute ($\delta_{I,I} \neq 0$), e.g. two different names.

Rule 2: Match With System Prompt

According to $\delta_{I,S}$, an interpretation that contains a correction of some item not occurring in the system's prompt is rejected. Example:

System: `Thus, you would like to talk to Mr. Jones?`

Best path:
`no not Johnson:name`
⇒ Inconsistent with the system prompt.
...
4th best path:
`no not Jones:name`
⇒ Consistent and selected.

Rule 3: Match With System's Belief

Interpretations which, after combination with the system's belief, do not refer to a valid database entry are also rejected according to $\delta_{I,S}$. For example:

System: `Which Martin would you like to talk to?`

Best path:
`Doctor:title Daves:name please`
db input: last name: Daves
first name: Martin
title: Doctor
⇒ No database match.
2nd best path:
`Doctor:title Davis:name please`
db input: last name: Davis
first name: Martin
title: Doctor
⇒ Consistent and selected.

4. EXPERIMENTAL RESULTS

A quantitative evaluation of the methods described above was carried out using field-test data from our PADIS prototype.

[3]The examples are translations of original German dialogues collected in the PADIS field test. Only the personal data (e.g. the names) have been exchanged for obvious reasons.

4.1. System Overview

The PADIS system provides telephone and room numbers, e-mail addresses, some private phone numbers, and direct call completion. It understands natural-language requests in fluently spoken German. A setup with a 500-entry database has successfully been field-tested in our research laboratory since early 1996. The vocabulary of the system contains around 1400 words, and the dialogue success rate is 90%.

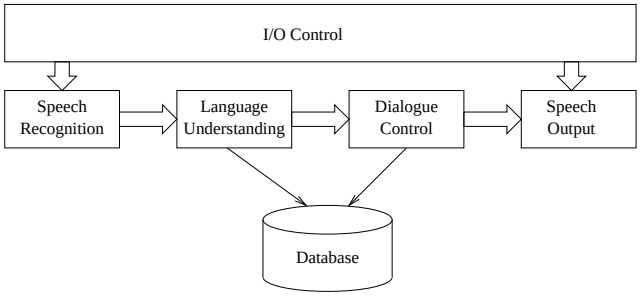

Figure 2: The PADIS signal-processing pipeline.

Figure 2 gives an overview of the system architecture. A more detailed description can be found in [8].

4.2. Results

We have evaluated the system performance for three setups:

- First, we used a traditional system without consistency checks (a simple first-best system, no N-best rescoring).
- In the next step, the first matching rule (consistency within utterance) and database knowledge were used.
- Finally, all matching rules described above were applied.

Table 2 contains the details of the corpora used for testing and grammar training. The acoustic models have been trained on 12.1h of speech data from our train schedule information system [5].

Table 2: PADIS field-test corpus characteristics.

	Training	Test
#Dialogues	2348 (4.5h)	1164 (1.5h)
#Turns	8214 (3.5 per dial.)	3198 (2.7 per dial.)
#Words	26445 (3.2 per turn)	8743 (2.7 per turn)

Table 3 gives the word error rate (WER), the attribute error rate (AER; measures substitutions, insertions, and deletions of information items), the average rank $\bar{n}$ of the selected hypothesis in the N-best list, and the test-set perplexity *PP*.

By applying the within-turn constraints, a relative reduction by 23% in attribute error rate, 15% in WER, and 35% in perplexity has been achieved. Since the lion share comes from rejecting hypotheses referring to non-existing database entries, the gain perceived by the users is much greater: Users turned out to be especially annoyed if a non-existing person is understood, because the computer 'could have known'.

Table 3: Results on the PADIS field-test corpus.

Setup	WER	AER	$\bar{n}$	*PP*
First best	28.9%	40.5%	1.0	25.1
Consistency within turn	24.6%	31.0%	3.2	16.2
Full model	24.4%	29.5%	3.8	–

The additional use of the system's belief and the current question has only led to a slight improvement on AER and has nearly had no effect on the WER. This is because the effects of the dialogue history inevitably remain small in dialogues consisting of just 2.7 turns an average. The minimum dialogue already takes two turns: "*How can I help you?* – Mr. Jones, please. – *You want to talk to Mr. Jones?* - Yes."

5. CONCLUSIONS AND OUTLOOK

We have presented an extended stochastic formulation of the speech understanding task that directly incorporates constraints from the inquiry system's database and the dialogue history. Using a 500-entry database, a total relative reduction of attribute errors by 27% and of word errors by 16% has been obtained. We believe that our method also enables us to build working systems with much larger databases.

Aside from extending the database, our future work will focus on model refinements. We expect the largest improvement from modelling the dependency of W and I on the current question's type, which is not yet regarded in eq. 2. Another topic will be adapting the a-priori probability of items that have been negated by the user in previous turns.

6. REFERENCES

1. M. Oerder and H. Aust. A realtime prototype of an automatic inquiry system. In *Proc. ICSLP94*, pp. 703–706, Yokohama, 1994.
2. M. Meteer and J.R. Rohlicek. Statistical language modelling combining N-gram and context-free grammars. In *Proc. ICASSP93*, Vol. II, pp. 37–40, Minneapolis, 1993.
3. R. Pieraccini and E. Levin. Stochastic representation of semantic structure for speech understanding. In *Proc. EUROSPEECH*, pp. 383–386, Genua, 1991.
4. E.P. Giachin. Phrase bigrams for continuous speech recognition. In *Proc. ICASSP95*, pp. 225–228, Detroit, 1995.
5. H. Aust, M. Oerder, F. Seide, and V. Steinbiss. Experience with the Philips automatic train timetable information system. In *Proc. IVTTA94*, pp. 67–72, Kyoto, 1994.
6. M. Oerder and H. Ney. Word graphs: An efficient interface between continuous-speech recognition and language understanding. In *Proc. ICASSP93*, Vol. II, pp. 119–122, Minneapolis, 1993.
7. B.-H. Tran, F. Seide, and V. Steinbiss. A word-graph based N-best search in continuous speech recognition. In *Proc. ICSLP96*, Philadelphia, 1996.
8. A. Kellner, B. Rueber, and F. Seide. A voice-controlled automatic telephone switchboard and directory information system. In *Proc. IVTTA96*, Basking Ridge, 1996.