BELIEF CONFIRMATION IN SPOKEN DIALOG SYSTEMS USING CONFIDENCE MEASURES

Christian Raymond †, Yannick Estève †, Frédéric Béchet †, Renato De Mori † , Géraldine Damnati ‡

† LIA, University of Avignon, BP1228, 84911 Avignon Cedex 9, France
‡ France Télécom R&D - DIH/IPS/RVA 2 av. Pierre Marzin 22307 Lannion Cedex 07, France

{*christian.raymond,yannick.esteve,frederic.bechet,renato.demori*}*@lia.univ-avignon.fr*
geraldine.damnati@rd.francetelecom.com

ABSTRACT

The approach proposed is an alternative to the traditional architecture of Spoken Dialogue Systems where the system belief is either not taken into account during the Automatic Speech Recognition process or included in the decoding process but never challenged. By representing all the conceptual structures handled by the Dialogue Manager by Finite State Machines and by building a conceptual model that contains all the possible interpretations of a given word-graph, we propose a decoding architecture that searches first for the best conceptual interpretation before looking for the best string of words. Once both *N*-best sets (at the concept level and at the word level) are generated, a verification process is performed on each N-best set using acoustic and linguistic confidence measures. A first selection strategy that does not include for the moment the Dialogue context is proposed and significant error reduction on the understanding measures are obtained.

1. INTRODUCTION

In a previous paper [1] a solution has been presented which integrates speech decoding with the generation of concept hypotheses. This solution adapts statistical language models (LMs) using expectations of concepts predicted by a system belief. A search engine is used which finds the best common path between the system knowledge represented by the composition of Stochastic Finite State Transducers (SFST) and the graph of word hypotheses generated by an Automatic Speech Recognition System (ASR).

The problem with this approach is that system belief may be incorrect, the language structures predicted by the belief can be incomplete and the user may not react to system prompts in a way that is consistent with system belief. It may happen that system belief boosts certain LM structures which force recognition of phrases which are consistent with belief but do not reproduce what the user said. An effective remedy to this problem consists in generating first the *N*-best conceptual interpretations of a user's utterance, then the *N*-best word strings for each interpretation.

The word strings are now a by-product of the understanding process: once an interpretation is chosen, the best or *n*-best word strings produced correspond to the best paths in the word graph (output by the ASR module) that can match the sequence of concepts corresponding to this interpretation. Driving the *n*-best string generation process by the Spoken Language Understanding module is an original aspect of this work.

Once both *N*-best sets (at the concept level and at the word level) are generated, a verification process is performed on each N-best set using confidence measures. The final decision on semantic interpretation is based on two kind of confidence measures: a purely acoustic one and a linguistic one.

Four situations can then occur:

- the interpretation selected corresponds to the system belief;
- another interpretation is chosen that corresponds to a context shift from the user;
- a refutation of the system belief is detected;
- no meaningful interpretation can be chosen and the utterance is rejected.

Experiments were carried out on a dialog corpus provided by France Télécom R&D for tourist inquiries. The test corpus contains 1274 sentences collected over the French telephone network in different days. The task has a vocabulary of 2200 words. A word graph is obtained for each spoken utterance and transformed into a Finite State Machine. Belief expectation is the union of three conceptual structures, namely, LOCATION, PRICE and FOOD_TYPE

and verification consists in assessing whether or not the expected concepts hypothesized by the system using its belief were present in the spoken sentence to be interpreted.

After presenting the theoretical framework of this work, we will describe the conceptual model used in order to turn the word graph into a conceptual transducer. Then we will present the confidence measures used to score the different concepts and propose a global architecture for Spoken Dialogue systems that implements these models. Finally, the results of a first evaluation on a corpus containing dialogues from a restaurant booking application are discussed.

2. CONCEPTUAL MODEL

Integrating semantic concept models into a statistical Language Model (LM) is not a novel idea [2, 3, 4]. However, most of these studies use conceptual *n*-gram models either to rescore a *n*-best list of hypothesis or to semantically tag the best string output by the ASR module.

In our approach, dialogue system belief is used to set up concept expectations with which LM is related and confidence measures are used to assess whether or not the user reacted accordingly to system belief. The role of the conceptual model is to estimate a set of global semantic interpretations of an utterance. These interpretations are made of strings of basic concepts that can be related to the dialogue itself or to the application domain. The dialogue state and the dialogue history can be used in order to choose among all the possible semantic interpretations.

Finding the best string of words for a given utterance corresponds to choose first the best interpretation for the utterance and then estimate the most probable word string corresponding to this interpretation.

A semantic interpretation is represented by a Finite State Transducer (FST) encoding regular grammars for each kind of concepts and filler models for the background text.

2.1. Statistical model

Let a dialogue system have a belief that generates expectations *B* about conceptual structures (or semantic interpretations). Expectation uncertainty is represented by a probability distribution *P(B)* which is non-zero for a set of conceptual structures expected at a given time. Thus for a general concept structure Γ and a description *Y* of the speech signal, one gets:

$$P(\Gamma \mid Y) = P(\Gamma, Y) = \sum_{B} P(\Gamma, Y, B)$$

with:

$$P(\Gamma, Y, B) = \sum_{W} P(W, \Gamma, Y, B)$$

and

$$P(W, \Gamma, Y, B) = P(Y \mid W) P(W, \Gamma, B)$$

$$P(\Gamma, W, B) = P(\Gamma \mid BW) P(W \mid B) P(B)$$

we obtain:

$$P(\Gamma \mid Y) = \sum_{B} \sum_{W} P(Y \mid W) P(\Gamma \mid BW) P(W \mid B) P(B) \quad (1)$$

Probability $P(\Gamma \mid BW)$ can be simply set equal to 0 for a conceptual structure which cannot be inferred from W. If the conceptual structure is part of the expectations of system beliefs and can be inferred unambiguously from W, then $P(\Gamma \mid BW)$ can be set to 1 as in many practical applications including the one considered in this paper.

Let Φ be the set of conceptual components, chunks of them or conceptual structures or semantic interpretations known to the system. Expectations derived from the system belief can be grouped into a set B1. Let B2 the complement of B1 w.r.t. Φ and F be a filler structure representing all the conceptual structures not in the application or just ignored by ignorance of the system knowledge. B1, B2 and F are the possible values for B in the formula (1) and their probabilities $P(B)$ can be established subjectively or by evaluating counts for user responses consistent with the belief, consistent with the application but not with the belief and inconsistent with the application knowledge.

Probability $P(W \mid B)$ is that of an LM which is adapted to the system belief.

Note that instead of considering all possible word strings in $\sum_W$, we limit the search space to the word graph output by the ASR module.

2.2. Conceptual entities

In this study, the conceptual entities are the units of the ontology which is the semantic knowledge of the Dialogue Manager. Their definitions rely on the dialogue strategy and they can be either related to dialogue management (confirmation, contestation, ...) or to the application domain (location, date, ...).

Section 5 presents results obtained when system belief predicts the most frequent application dependent concepts, namely: *LOCATION*, *PRICE* and *FOOD_TYPE*. They can be described as follows:

- *LOCATION*: an expression related to a restaurant location (e.g. *a restaurant near Bastille*);
- *PRICE*: the price range of a restaurant (e.g. *around one hundred euros*);
- *FOOD_TYPE*: the kind of food requested by the caller (e.g. *an Indian restaurant*).

These entities are expressed in the training corpus by short sequences of words containing three kinds of token: headwords like *Bastille*, concept related words like *restaurant* and modifier tokens like *near*. In order to automatically learn regular grammars specific to each concept, the training corpus is manually tagged at the concept level. Regular grammars are induced from these examples by generalizing some of their tokens by means of syntactic and semantic criteria.

2.3. Words to Concepts Transducer

Each concept C_k of the dialogue application is associated with a regular grammar. These grammars are represented by Finite State Machines called *acceptors* (A_k for the concept C_k). In order to process strings of words that don't belong to any concept, a filler model, called A_F is used. Because the same string of words can't belong to both a concept model and the background text, all the paths contained in the acceptors A_k are removed from the filler model A_F in the following way:

$$A_F = \Sigma * - \bigcup_{k=1}^{m} A_k$$

where Σ is the word lexicon of the application and m is the number of concepts used.

All these acceptors are now turned into transducers that take words as input symbols and *start* or *end* concept tags as output symbols. Indeed, all acceptors A_k become transducers T_k where the first transition emits the symbol $< C_k >$ and the last transition the symbol $< /C_k >$. Similarly the filler model becomes the transducer T_{bk} which emits the symbols $< BCK >$ and $< /BCK >$. Except these start and end tags, no other symbols are emitted: all words in the concept or background transducers emit an *epsilon* symbol.

Finally all these transducers are linked together in a single model called $T_{concept}$ as presented in figure 1.

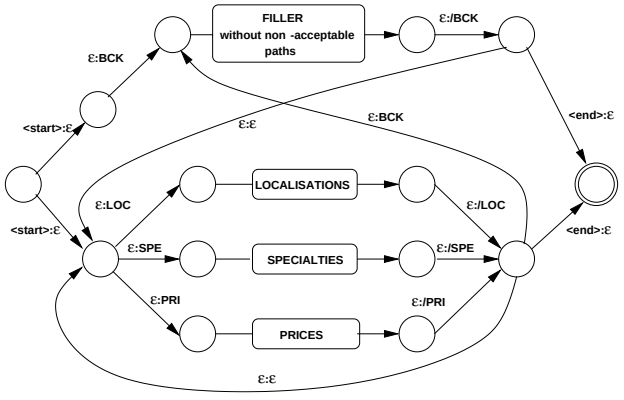

Fig. 1. Words-to-Concepts Transducer

This word-to-concept transducer $T_{Concept}$ associates only one sequence of concepts (*output label paths*) to a given sequence of words (*input label path*): only one segmentation of a sequence of words can be proposed for a given sequence of concepts. Weight associated to emission of a concept are probabilities $P(B)$ defined in section 2.1 which can evolve according to the dialogue state. These probabilities represent the system belief.

3. CONFIDENCE MEASURES

3.1. Linguistic confidence measure

In many practical applications, the training data available is biased by the fact that the corpus has been collected with a limited number of speakers and in a limited time period. It results in a limited amount of training data available. Then many n-grams, that would appear more than once in an ideally large training corpus, have a probability computed with a back-off model, which is a cause of many recognition errors. In order to assess the impact of the absence of observed trigrams as a potential cause of recognition errors, an LM consistency measure is introduced. Let us indicate as J_0 the set of 3-grams observed at least once in the training set and J_U the set of 3-grams occurring in an utterance *U*. An LM-based confidence measure *CONS(LM)* is introduced, defined as follows:

$$CONS(LM) = \frac{\mid J_0 \wedge J_U \mid}{\mid J_U \mid} \quad (2)$$

The definition of *CONS(LM)* is inspired by measures proposed in [5]. Its computation is very fast and it could be performed over an entire sentence or only on a sub-sentence that represents conceptual information.

3.2. Acoustic confidence measure

As the previously defined conceptual word strings identification is mainly semantically and linguistically driven, we have chosen to apply a purely acoustic confidence measure. Actually, this information is believed to be well suited as it is complementary to the previous approach. The confidence measure relies on the comparison of the acoustic likelihood provided by the speech recognition model for a given hypothesis to the one that would be provided by a totally unconstrained phoneme loop model. In order to be consistent with the general model, the acoustic units are kept identical and the loop is over context dependent phonemes, namely allophones [6].

3.2.1. Likelihood ratio at the word level

For a hypothesis *W* identified by the general model (λ_G) from frame t_0 to frame t_n, the likelihood of the speech sig-

nal *Y* is compared to the likelihood of the same portion of signal over the unconstrained allophone loop. The likelihood ratio is defined as follows:

$$LR(Y \mid W) = \frac{P(Y \mid \lambda_G)}{P(Y \mid \lambda_{loop})} \quad (3)$$

In order to be able to compare the ratio values for different words, we actually compare the log-likelihood and normalize the difference by the number of frames over which it is computed.

$$\Delta_{loop}(Y \mid W) = \frac{logLR(Y \mid W)}{N_{frame}(W)}$$
$$= \frac{1}{N_{frame}(W)}[logP(Y \mid \lambda_G) - log \quad P(Y \mid \lambda_{loop})] \quad (4)$$

Practically, $\Delta loop(Y \mid W)$ is computed from the raw acoustic score provided during the speech recognition decoding process and :

$$\Delta_{loop}(Y \mid W) = \frac{1}{N_{frame}(W)}[Sc_G(Y) - Sc_{loop}(Y)] \quad (5)$$

Due to the lack of constraints, the likelihood of the speech signal overλ_{loop} is higher than the likelihood over λ_G. It can be viewed as an upper bound for $P(Y \mid \lambda_G)$. Thus, $\Delta_{loop}(Y|W)$ is a negative value that is to be interpreted as follows: the closer to zero the more reliable is the hypothesis *W* for *Y*.

3.2.2. Likelihood ratio at the concept level

In order to score the different concept hypothesis, the previous confidence measure easily extends to the concept level. In fact, for simplicity purpose, the Δ_{loop} for a word string hypothesis is derived from the Δ_{loop} of each word component. Let Γ be a conceptual structure composed of *n* words $W_1, ...W_n$, $\Delta_{loop}(Y \mid \Gamma)$ is approximated by :

$$\Delta_{loop}(Y \mid \Gamma) = \frac{1}{\sum_{i=1}^{n} N_{frame}(W_i)} \times \sum_{i=1}^{n} N_{frame}(W_i)\Delta_{loop}(Y_i \mid W_i) \quad (6)$$

4. GENERAL ARCHITECTURE

For the implementation of the Finite State Machines presented in section 2.3 we used the AT&T FSM Library [7] which provides very efficient data format and algorithms for managing both acceptors and transducers. We are now going to present the different steps in the process that uses the conceptual model in order to give to the dialogue manager a very limited amount of hypotheses which are semantically different and validated by confidence measures.

4.1. ASR word graph with conceptual information

A first decoding process using acoustic and linguistic statistical models is performed. This generates a word graph which is turned into a stochastic transducer T_{ASR}. The weights associated to the transitions in this transducer are combinations of acoustic and linguistic scores. Input and output labels (which are words of the lexicon) are identical. By performing a *composition* operation between T_{ASR} and the conceptual model $T_{Concept}$ we obtain new transducer T_{Decod} integrating semantic and belief knowledge to the ASR word graph:

$$T_{Decod} = T_{ASR} \circ T_{Concept}$$

In T_{Decod} the input labels are words and the output labels are conceptual tags.

4.2. *N*-best list of conceptual interpretations

The transducer T_{Decod} is converted into an acceptor by projection on its output labels. This projection creates an acceptor whose label paths are all the sequences of concepts that can be emitted from T_{Decod}. By applying an *n*-best search algorithm on this acceptor, one obtains the *n*-best list of interpretations (i.e. sequence of concepts), which can be found in the word graph. In practice, all the sequences of concepts that can be emitted by T_{Decod} can be found with a limited size of *n*-best interpretations [1]. Note that the score associated to a sequence of concepts is the sum of the probabilities of all the sequences of words from T_{Decod} which can emit this sequence.

4.3. *N*-best list of word strings

Each sequence of concepts *i* previously obtained is now turned into a transducer T_{inter_i} with identical input and output symbols: the *start* and *end* tags of each concept. By processing a *composition* operation between the transducer T_{Decod} and a transducer T_{inter_i}, one obtains all the possible word paths in the graph that can emit the sequences of concepts corresponding to the interpretation *i*.

A *N*-best list of word strings for each interpretation *i* is simply obtained by performing a search algorithm on the transducer $T_{Decod} \circ T_{inter_i}$.

Figure 2 shows an architecture diagram of the software modules performing the above described operations.

4.4. Selection strategy with confidence measures

In a final stage, the confidence score presented in 3 are used to re-score the *n*-best hypotheses of each utterance interpretation, providing additional information to the Dialogue Manager for choosing between the different hypotheses.

[1] this depends of the number of concepts used by the application and of the size of the words graph generated in the first decoding pass

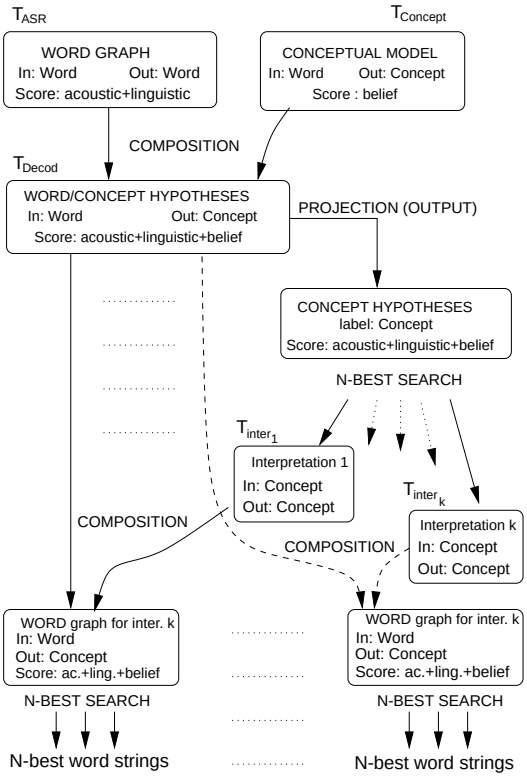

Fig. 2. General Architecture

However, for evaluation purpose, we have tried to apply a threshold on the values of $\Delta_{loop}(Y \mid \Gamma)$ and the linguistic confidence score and to simply reject concept hypothesis that are below these thresholds.

As a matter of fact, most of the errors made at the conceptual level on the one-best output of the system are insertions of concepts due to out-of-vocabulary words or speech repairs. Rejecting wrong hypotheses should therefore mainly reduce the insertion rate, which is crucial in a spoken dialogue context.

This simple selection strategy can be described as follows:

- the conceptual interpretations are sorted according to their scores as presented in 4.2 and only the best word string for each interpretation is kept;
- all the hypothesis that have at least one of their confidence measures below a given threshold are suppressed.

If no hypotheses remain after this filtering process, one can either reject the utterance or use as back off the one-best hypothesis for evaluation purpose.

5. EVALUATION

Experiments were carried out on a dialog corpus provided by France Télécom R&D for tourist inquiries. The development corpus contains 2.2K sentences and the test corpus 1.7K sentences collected over the French telephone network in different days. The task has a vocabulary of 2200 words. All the thresholds for the selection strategy have been tuned on the development corpus and the results are given on the test corpus.

Because this was an out-of-context evaluation (no information about the dialogue context was attached to the utterances to process), we arbitrary set the belief of the Dialog Manager (DM) to expect any string of concepts containing the following items: *LOCATION*, *PRICE* and *FOOD_TYPE*.

A first experiment has the purpose of comparing the ability of the proposed architecture in recognizing concepts (with their values), with that of a baseline architecture, which generates sequences of word hypotheses with a general LM with no conceptual models.

Two measures are considered: Concept Error Rate (*CER*) and Understanding Error Rate (*UER*). *CER* is related to the concept tags alone and *UER* is related to the normalized values of the concepts detected. These values are obtained by a set of rules that translate the word strings detected as concepts into tokens representing the values. To each concept is associated a single value. For example, to the context: *a restaurant near Bastille* is associated the value: *BASTILLE*.

The reference corpus is made by filtering the manual transcriptions of the test corpus in order to keep only the concept tags and values for each utterance. The utterances containing no concepts as considered empty but they are kept in the reference corpus in order to score false insertions. 28% of the utterances of the reference corpus contain at least one concept.

With Γ as the concept structure of each utterance of the test corpus, *CER* and *UER* are defined as follows:

$$CER = \frac{S_c + D_c + I_c}{T} \times 100$$

$$UER = \frac{S_v + D_c + I_c}{T} \times 100$$

where S_c indicates the substitution of an attribute of Γ, S_v the substitution of a concept value, D_c indicates deletion of an attribute and I_c indicates insertion. T is the total number of concepts in the reference test file.

A lower bound of *CER* and *UER* is obtained by searching in the graph of word hypotheses the exact structure of Γ present in the spoken utterance. A value of 8% was found for *CER* and 12.6% for *UER*. By generating the *N*-best word hypotheses with the baseline system and our method, we can compare the lowest error rate that could be obtained if

we knew how to choose the best hypotheses among the *N* ones. Table 1 summarized these results for *N*=3, *N*=6 and *N*=9. The values *CER*=8.4% and *UER*=16.8% were found for *N*=3 using the architecture whose scheme is shown in Figure 2, while, for the baseline, *CER*=15.2 and *UER*=20.2 for *N*=3. This allows one to conclude that the proposed architecture is more precise than the baseline in generating concept hypotheses.

Error rate	**baseline**			**new arch.**		
N	**3**	**6**	**9**	**3**	**6**	**9**
CER %	15.2	11.7	9.6	8.4	8.4	8.4
UER %	20.2	16.6	14.1	16.8	14.0	13.4

Table 1. Lowest error rates (Concept Error Rate and Understanding Error Rate) obtained by considering the N-best hypotheses (with N=3,6 and 9) output by the baseline system and the new architecture

Table 2 shows the results obtained by considering only the 1-best hypothesis produced by the baseline system (*baseline*) and the best hypothesis among the 3-best ones presented in table 1 with the filtering process described in section 4 (*new arch.)*. One can see that even if a significant improvement is obtained on the understanding measures *CER* and *UER*, these results are still far from the error rate lowest bounds obtained with the 3-best list of hypotheses presented in table 1. A closer look to these results shows that it's the insertion error rate that decreases most between the baseline system and the new architecture with the filtering process (from 13.8% to 8.7% for the *UER*). This result is very relevant in a dialogue context as every inserted concept in the understanding process of an utterance might lead the dialogue manager in a wrong path. However, these results suggest that the decision strategy for selecting the correct structure of Γ with the correct values among the top 3 hypotheses has still to be investigated. New decision strategies, based on a better combination of the different confidence measures as well as the integration of dialogue context information are actually under investigation.

Error rate	**baseline**	**new arch.**
WER %	29.9	29.6
CER %	26.0	23.7
UER %	32.5	29.3

Table 2. Error rates (Word Error Rate, Concept Error Rate and Understanding Error Rate) on the 1-best hypothesis for the baseline system and the new architecture

6. CONCLUSION

The approach proposed is an alternative to the traditional sequential architecture of Spoken Dialogue Systems where the system belief is either not taken into account during the Automatic Speech Recognition process or included in the decoding process but never challenged. By representing all the conceptual structures handled by the DM by Finite State Machines and by building a conceptual model that contains all the possible interpretations, we propose a decoding architecture that search first for the best conceptual interpretation before looking for the best string of words. Confidence measures, both on acoustic and linguistic features are also used in order to filter the *N*-best lists produced. A first selection strategy that does not include for the moment the Dialogue context is proposed and significant error reduction on the understanding measures are obtained.

7. REFERENCES

[1] Yannick Estève, Christian Raymond, Frédéric Béchet, and Renato De Mori, "Conceptual decoding for spoken dialog systems," in *Proceedings of European Conference on Speech Communication and Technology (Eurospeech)*, Geneva, Switzerland, 2003.

[2] Pieraccini R., Levin E., and Lee C.-H, "Stochastic representation of conceptual structure in the atis task," in *Speech and Natural Language Work-shop*, Morgan Kaufmann publ, Los Altos, CA, 1991, pp. 121–124.

[3] Y.Y. Wang and A. Acero, "Concept acquisition in example based grammar authoring," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Hong Kong, 2003, pp. 284–287.

[4] Y. He and S. Young, "Hidden vector state model for hierarchical semantic parsing," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Hong Kong, 2003, pp. 268–271.

[5] R. De Mori Y. Estève, C. Raymond and D. Janiszek, "On the use of linguistic consistency in systems for human-computer dialogs," *IEEE Transactions on Speech and Audio Processing*, vol. (Accepted for publication, in press), 2003.

[6] Katarina Bartkova and Denis Jouvet, "Modelization of allophones in a speech recognition system," in *Proceedings of International Conference of Phonetic Science*, Aix-en-Provence, France, 1991.

[7] Mehryar Mohri, Fernando Pereira, and Michael Riley, "Weighted finite-state transducers in speech recognition," *Computer, Speech and Language*, vol. 16, no. 1, pp. 69–88, 2002.