On the use of confidence for statistical decision in dialogue strategies

Christian Raymond[1] **Frédéric Béchet**[1] **Renato de Mori**[1] **Géraldine Damnati**[2]
[1] LIA/CNRS, University of Avignon France [2] France Telecom R&D, Lannion, France
christian.raymond,frederic.bechet,renato.demori@lia.univ-avignon.fr
geraldine.damnati@rd.francetelecom.com

Abstract

This paper describes an interpretation and decision strategy that minimizes interpretation errors and perform dialogue actions which may not depend on the hypothesized concepts only, but also on confidence of what has been recognized. The concepts introduced here are applied in a system which integrates language and interpretation models into Stochastic Finite State Transducers (SFST). Furthermore, acoustic, linguistic and semantic confidence measures on the hypothesized word sequences are made available to the dialogue strategy. By evaluating predicates related to these confidence measures, a decision tree automatically learn a decision strategy for rescoring a n-best list of candidates representing a user's utterance. The different actions that can be then performed are chosen according to the confidence scores given by the tree.

1 Introduction

There is a wide consensus in the scientific community that human-computer dialogue systems based on spoken natural language make mistakes because the Automatic Speech Recognition (ASR) component may not hypothesize some of the pronounced words and the various levels of knowledge used for recognizing and reasoning about conceptual entities are imprecise and incomplete. In spite of these problems, it is possible to make useful applications with dialogue systems using spoken input if suitable interpretation and decision strategies are conceived that minimize interpretation errors and perform dialogue actions which may not depend on the hypothesized concepts only, but also on confidence of what has been recognized.

This paper introduces some concepts developed for telephone applications in the framework of stochastic models for interpretation and dialogue strategies, a good overview of which can be found in (Young, 2002).

The concepts introduced here are applied in a system which integrates language and interpretation models into Stochastic Finite State Transducers (SFST). Furthermore, acoustic, linguistic and semantic confidence measures on the hypothesized word sequences are made available to the dialogue strategy. A new way of using them in the dialogue decision process is proposed in this paper.

Most of the Spoken language Understanding Systems (SLU) use semantic grammars with semantic tags as non-terminals (He and Young, 2003) with rules for rewriting them into strings of words.

The SFSTs of the system used for the experiments described here, represent knowledge for the basic building blocks of a frame-based semantic grammar. Each block represents a property/value relation. Different SFSTs may share words in the same sentence. Property/value hypotheses are generated with an approach described in (Raymond et al., 2003) and are combined into a sentence interpretation hypothesis in which the same word may contribute to more than one property/value pair. The dialogue strategy has to evaluate the probability that each component of each pair has been correctly hypothesized in order to decide to perform an action that minimizes the risk of user dissatisfaction.

2 Overview of the decoding process

The starting point for decoding is a lattice of word hypotheses generated with an n-gram language model (LM). Decoding is a search process which detects combinations of specialized SFSTs and the n-gram LM. The output of the decoding process consists of a n-best list of *conceptual interpretations* Γ. An interpretation Γ is a set of property/value pairs $s_j = (c_j, v_j)$ called *concepts*. c_j is the concept tag and v_j is the concept value

of s_j. Each concept tag c_j is represented by a SFST and can be related either to the dialogue application (phone number, date, location expression, etc.) or to the dialogue management (confirmation, contestation, etc.). To each string of words recognized by a given SFST c_j is associated a value v_j representing a normalized value for the concept detected. For example, to the word phrase: `on July the fourteenth`, detected by a SFST dedicated to process dates, is associated the value: `????/07/14`.

The n-best list of interpretations output by the decoding process is structured according to the different concept tag strings that can be found in the word lattice. To each concept tag string is attached another n-best list on the concept values. This whole n-best is called a *structured n-best*. After presenting the statistical model used in this study, we will describe the implementation of this decoding process.

3 Statistical model

The contribution of a sequence of words W to a conceptual structure Γ is evaluated by the posterior probability $P(\Gamma \mid Y)$, where Y is the description of acoustic features. Such a probability is computed as follows:

$$P(\Gamma \mid Y) = \frac{\sum_{W \in SW} P(Y \mid W) P(\Gamma \mid W)^{\delta} P(W)^{\lambda}}{\sum_{W \in SW} P(Y \mid W) P(W)^{\lambda}} \quad (1)$$

where $P(Y \mid W)$ is provided by the acoustic models, $P(W)$ is computed with the LM. Exponents δ and λ are respectively a semantic and a syntactic fudge factor. SW corresponds to the set of word strings that can be found in the word lattice. $P(\Gamma \mid W)$ is computed by considering that thus:

$$P(\Gamma \mid W) = P(s_1 \mid W). \prod_{j=2}^{J} P(s_j \mid s_1^{j-1} W) \quad (2)$$
$$P(s_j \mid s_1^{j-1} W) \approx P(s_j \mid W)$$

If the conceptual component s_j is hypothesized with a sentence pattern $\pi_j(W)$ recognized in W and $\pi_k(W)$ triggers a pair s_k and there is a training set with which the probabilities $P(\pi_k(W) \mid s_k)\ \forall k$, can be estimated, then the posterior probability can be obtained as follows:

$$P(s_j \mid W) = \frac{P(\pi_j(W) \mid s_j) P(s_j)}{\sum_{k=1}^{K} P(\pi_k(W) \mid s_k) P(s_k)} \quad (3)$$

where $P(s_k)$ is a unigram probability of conceptual components.

4 Structured N-best list

N-best lists are generally produced by simply enumerating the *n* best paths in the word graphs produced by Automatic Speech Recognition (ASR) engines. The scores used in such graphs are usually only a combination of acoustic and language model scores, and no other linguistic levels are involved. When an n-best word hypothesis list is generated, the differences between the hypothesis *i* and the hypothesis *i+1* are often very small, made of only one or a few words. This phenomenon is aggravated when the ASR word graph contains a low confidence area, due for example to an Out-Of-Vocabulary word, to a noisy input or to a speech disfluency.

This is the main weakness of this approach in a Spoken Dialogue context: not all words are important to the Dialogue Manager, and all the n-best word hypotheses that differ only between each other because of some speech disfluency effects can be considered as equals. That's why it is important to generate not only a n-best list of word hypotheses but rather a n-best list of interpretations, each of them corresponding to a different meaning from the Dialogue Manager point of view.

We propose here a method for directly extracting such a structured n-best from a word lattice output by an ASR engine. This method relies on operations between Finite State Machines and is implemented thanks to the AT&T FSM toolkit (see (Mohri et al., 2002) for more details).

4.1 Word-to-Concept transducer

Each concept c_k of the dialogue application is associated with an FSM. These FMSs are 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 `<Ck>` and the last transition the symbol `</Ck>`. 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 empty symbol.

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

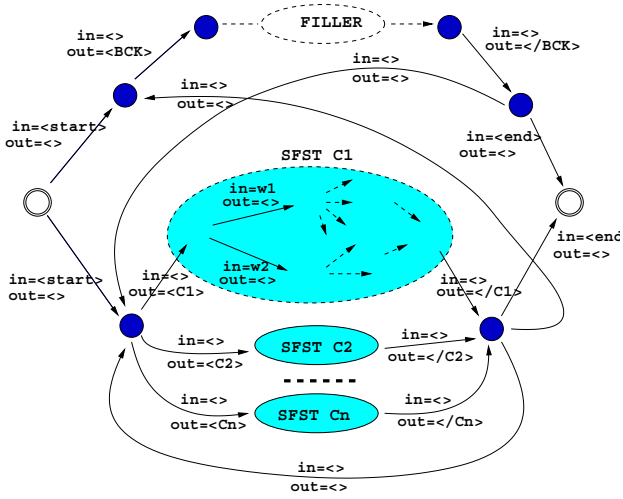

Figure 1: Word-to-Concept Transducer

4.2 Processing the ASR word lattice

The ASR word lattice is coded by an FSM: an acceptor L where each transition emits a word. The cost function for a transition corresponds to the acoustic score of the word emitted.

The first step in the word lattice processing consists of rescoring each transition of L by means of a 3-gram Language Model (LM) in order to obtain the probabilities $P(W)$ of equation 1. This is done by composing the word lattice with a 3-gram LM also coded as an FSM (see (Allauzen et al., 2003) for more details about statistical LMs and FSMs).

The resulting FSM is then composed with the transducer $T_{Concept}$ in order to obtain the word-to-concept transducer L'. A path in L' corresponds to a word string if only the input symbols of the transducer are considered and its score is the one expressed by equation 1; similarly by considering only the output symbols, a path in L' corresponds to a concept tag string.

The structured n-best list is directly obtained from L': by extracting the n-best concept tag strings (output label paths) we obtain an n-best list *on the conceptual interpretations*. The score of each conceptual interpretation is the sum of all the word strings (input label paths) in the word lattice producing the same interpretation.

Finally, for every conceptual interpretations C kept at the previous step, a local n-best list on the word strings is calculated by selecting in L' the best paths outputting the string C.

The resulting structured n-best is illustrated by the following example. If we keep the 2 best conceptual interpretations C_1, C_2 of a transducer L' and, for each of these, the 2 best word strings, we obtain:

```
1 : C1 = <c1_1,c1_2,..,c1_x>
  1.1 : W1.1 = <v1.1_1,v1.1_2,..,v1.1_x>
  1.2 : W1.2 = <v1.2_1,v1.2_2,..,v1.2_x>
2 : C2 = <c2_1,c2_2,..,c2_y>
  2.1 : W2.1 = <v2.1_1,v2.1_2,..,v2.1_y>
  2.2 : W2.2 = <v2.2_1,v2.2_2,..,v2.2_y>
```

where `<ci_1,ci_2,..,ci_y>` is the conceptual interpretation at the rank `i` in the n-best list; `Wi.j` is the word string ranked `j` of interpretation `i`; and `vi.j_k` is the concept value of the k^{th} concept `ci_k` of the j^{th} word string of interpretation `i`.

5 Use of correctness probabilities

In order to select a particular interpretation Γ (conceptual interpretation + concept values) from the structured n-best list, we are now interested in computing the probability that Γ is correct, given a set of confidence measures M: $P(\Gamma \mid M)$. The choice of the confidence measures determines the quality of the decision strategy. Those used in this study are briefly presented in the next sections.

5.1 Confidence measures

5.1.1 Acoustic confidence measure (AC)

This 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. This confidence measure is used at the utterance level and at the concept level (see (Raymond et al., 2003) for more details).

5.1.2 Linguistic confidence measure (LC)

In order to assess the impact of the absence of observed trigrams as a potential cause of recognition errors, a Language Model consistency measure is introduced. This measure is simply, for a given word string candidate, the ratio between the number of trigrams observed in the training corpus of the Language Model *vs.* the total number of trigrams in the same word string. Its computation is very fast and the confidence scores obtained from it give interesting results as presented in (Estève et al., 2003).

5.1.3 Semantic confidence measure (SC)

Several studies have shown that text classification tools (like Support Vector Machines or Boosting algorithms) can be an efficient way of labeling an utterance transcription with a semantic label such as a call-type (Haffner et al., 2003) in a Spoken Dialogue context. In our case, the semantic labels attached to an utterance are the different

concepts handled by the Dialogue Manager. One classifier is trained for each concept tag in the following way:

Each utterance of a training corpus is labeled with a tag, manually checked, indicating if a given concept occurs or not in the utterance. In order to let the classifier model the context of occurrence of a concept rather than its value we removed most of the concept headwords from the list of criterion used by the classifier.

During the decision process, if the interpretation evaluated contains 2 concepts c_1 and c_2, then the classifiers corresponding to c_1 and c_2 are used to give to the utterance a confidence score of containing these two concepts.

The text classifier used in the experimental section is a decision-tree classifier based on the Semantic-Classification-Trees introduced for the ATIS task by (Kuhn and Mori, 1995) and used for semantic disambiguation in (Béchet et al., 2000).

5.1.4 Rank confidence measure (R)

To the previous confidence measures we added the rank of each candidate in its n-best. This rank contains two numbers: the rank of the interpretation of the utterance and the rank of the utterance among those having the same interpretation.

5.2 Decision Tree based strategy

As the dependencies of these measures are difficult to establish, their values are transformed into symbols by vector quantization (VQ) and conjunctions of these symbols expressing relevant statistical dependencies are obtained by a decision tree which is trained with a development set of examples. At the leaves probabilities $P(M|\Gamma)$ are obtained when Γ represents any correct hypothesis, the case in which only the properties have been correctly recognized or both properties and values have errors. With these probabilities we are now able to estimate $P(\Gamma \mid M)$ in the following way:

$$P(\Gamma \mid M) = \frac{1}{1 + \frac{P(M|\Gamma')P(\Gamma')}{P(M|\Gamma)P(\Gamma)}} \quad (4)$$

where Γ' indicates that the interpretation in question is incorrect and $P(M|\Gamma') = 1 - P(M|\Gamma)$.

6 From hypotheses to actions

Once concepts have been hypothesized, a dialog system has to decide what action to perform. Let $A = a_j$ be the set of actions a system can perform. Some of them can be requests for clarification or repetition. In particular, the system may request the repetition of the entire utterance. Performing an action has a certain risk and the decision about the action to perform has to be the one that minimizes the risk of user dissatisfaction.

It is thus possible that some or all the hypothesized components of a conceptual structure Γ do not correspond to the user intention because the word sequence W based on which the conceptual hypothesis has been generated contains some errors. In particular, there are requests for clarification or repetition which should be performed right after the interpretation of an utterance in order to reduce the stress of the user. It is important to notice that actions consisting in requests for clarification or repetition mostly depend on the probability that the interpretation of an utterance is correct, rather than on the utterance interpretation.

The decoding process described in section 2 provides a number of hypotheses containing a variable number of pairs $s_j = (c_j, v_j)$ based on the score expressed by equation 1.

$P(\Gamma \mid M)$ is then computed for these hypotheses. The results can be used to decide to accept an interpretation or to formulate a clarification question which may imply more hypotheses.

For simplification purpose, we are going to consider here only two actions: accepting the hypothesis with the higher $P(\Gamma \mid M)$ or rejecting it. The risk associated to the acceptation decision is called ρ_{fa} and corresponds to the cost of a false acceptation of an incorrect interpretation. Similarly the risk associated to the rejection decision is called ρ_{fr} and corresponds to the cost of a false rejection of a correct interpretation. In a spoken dialogue context, ρ_{fa} is supposed to be higher than ρ_{fr}.

The choice of the action to perform is determined by a threshold δ on $P(\Gamma \mid M)$. This threshold is tuned on a development corpus by minimizing the total risk R expressed as follows:

$$R = \rho_{fa} \times \frac{N_{fa}}{N_{total}} + \rho_{fr} \times \frac{N_{fr}}{N_{total}} \quad (5)$$

N_{fa} and N_{fr} are the numbers of false acceptation and false rejection decisions on the development corpus for a given value of δ. N_{total} is the total number of examples available for tuning the strategy.

The final goal of the strategy is to make negligible N_{fa} and the best set of confidence measures is the one that minimizes N_{fr}. In fact, the cost of these cases is lower because the corresponding action has to be a request for repetition.

Instead of simply discarding an utterance if $P(\Gamma \mid M)$ is below δ, another strategy we are investigating consists of estimating the probability that the conceptual interpretation alone (without the concept values) is correct. This probability can be estimated the same way as $P(\Gamma \mid M)$ and can be used to choose a third kind of actions: accepting the conceptual meaning of an utterance but asking for clarifications about the values of the concepts.

A final decision about the strategy to be adopted should

be based on statistics on system performance to be collected and updated after deploying the system on the telephone network.

7 Experiments

7.1 Application domain

The application domain considered in this study is a restaurant booking application developed at France Telecom R&D. At the moment, we only consider in our strategy the most frequent concepts related to the application domain: *PLACE*, *PRICE* and *FOOD_TYPE*. They can be described as follows:

- *PLACE*: an expression related to a restaurant location (eg. *a restaurant near Bastille*);
- *PRICE*: the price range of a restaurant (eg. *less than a hundred euros*);
- *FOOD_TYPE*: the kind of food requested by the caller (eg. *an Indian restaurant*).

These entities are expressed in the training corpus by short sequences of words containing three kinds of token: head-words like *Bastille*, concept related words like *restaurant* and modifier tokens like *near*.

A single value is associated to each concept entity simply be adding together the head-words and some modifier tokens. For example, the values associated to the three contexts presented above are: `Bastille` , `less+hundred+euros` and `indian`.

In the results section a concept detected is considered a success only if the tag exists in the reference corpus and if both values are identical. It's a binary decision process: a concept can be considered as a false detection even if the concept tag is correct and if the value is partially correct. The measure on the errors (insertion, substitution, deletion) of these concept/value tokens is called in this paper the *Understanding Error Rate*, by opposition to the standard Word Error Rate measure where all words are considered equals.

7.2 Experimental setup

Experiments were carried out on a dialogue corpus provided by France Telecom R&D. The task has a vocabulary of 2200 words. The language model used is made of 44K words. For this study we selected utterances corresponding to answers to a prompt asking for the kind of restaurant the users were looking for. This corpus has been cut in two: a development corpus containing 511 utterances and a test corpus containing 419 utterances. This development corpus has been used to train the decision tree presented in section 5.2. The Word Error Rate on the test corpus is 22.7%.

7.3 Evaluation of the rescoring strategy

Table 1 shows the results obtained with a rescoring strategy that selects, from the structured n-best list, the hypothesis with the highest $P(\Gamma \mid M)$. The baseline results are obtained with a standard maximum-likelihood approach choosing the hypothesis maximizing the probability $P(\Gamma \mid Y)$ of equation 1. No rejection is performed in this experiment.

The size of the n-best lists was set to 12 items: the first 4 candidates of the first 3 interpretations in the structured n-best list. The gain obtained after rescoring is very significant and justify our 2-step approach that first extract an n-best list of interpretations thanks to $P(\Gamma \mid Y)$ and then choose the one with the highest confidence according to a large set of confidence measures M. This gain can be compared to the one obtained on the Word Error Rate measure: the WER drops from 21.6% to 20.7% after rescoring on the development corpus and from 22.7% to 22.5% on the test corpus. It is clear here that the WER measure is not an adequate measure in a Spoken Dialogue context as a big reduction in the Understanding Error Rate might have very little effect on the Word Error Rate.

Corpus	baseline	rescoring	UER reduction %
Devt.	15.0	12.4	17.3%
Test	17.7	14.5	18%

Table 1: Understanding Error Rate results with and without rescoring on structured n-best lists (n=12) (no rejection)

7.4 Evaluation of the decision strategy

In this experiment we evaluate the decision strategy consisting of accepting or rejecting an hypothesis Γ thanks to a threshold on the probability $P(\Gamma \mid M)$. Figure 2 shows the curve UER *vs.* utterance rejection on the development and test corpora. As we can see very significant improvements can be achieved with very little utterance rejection. For example, at a 5% utterance rejection operating point, the UER on the development corpus drops from 15.0% to 8.6% (42.6% relative improvement) and from 17.7% to 11.4% (35.6% relative improvement).

By using equation 5 for finding the operating point minimizing the risk fonction (with a cost $\rho_{fa} = 1.5 \times \rho_{fr}$) on the development corpus we obtain:

- on the development corpus: UER=6.5 utterance rejection=13.1
- on the test corpus: UER=9.6 utterance rejection=15.9

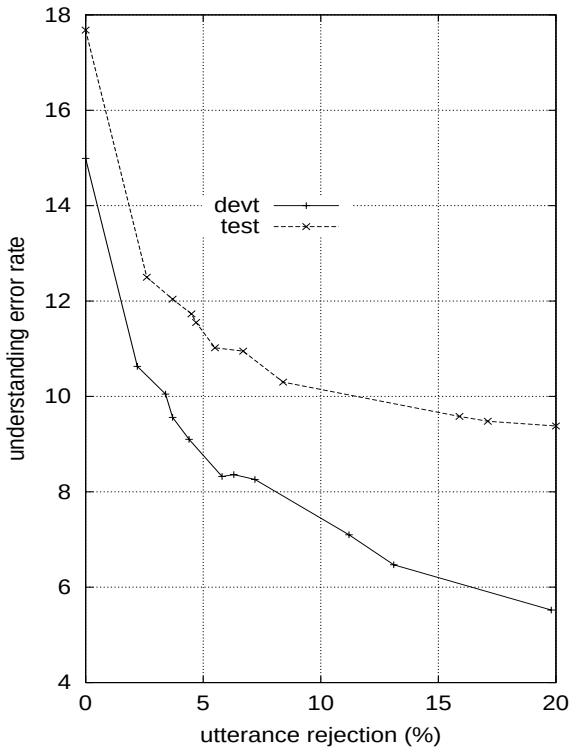

Figure 2: Understanding Error Rate *vs.* utterance rejection on the development and test corpora

8 Conclusion

This paper describes an interpretation and decision strategy that minimizes interpretation errors and perform dialogue actions which may not depend on the hypothesized concepts only, but also on confidence of what has been recognized. The first step in the process consists of generating a structured n-best list of conceptual interpretations of an utterance. A set of confidence measures is then used in order to rescore the n-best list thanks to a decision tree approach. Significant gains in Understanding Error Rate are achieved with this rescoring method (18% relative improvement). The confidence score given by the tree can also be used in a decision strategy about the action to perform. By using this score, significant improvements in UER can be achieved with very little utterance rejection. For example, at a 5% utterance rejection operating point, the UER on the development corpus drops from 15.0% to 8.6% (42.6% relative improvement) and from 17.7% to 11.4% (35.6% relative improvement). Finally the operating point for a deployed dialogue system can be chosen by explicitly minimizing a risk function on a development corpus.

References

Cyril Allauzen, Mehryar Mohri, and Brian Roark. 2003. Generalized algorithms for constructing statistical language models. In *41st Annual Meeting of the Association for Computational Linguistics (ACL'03), Sapporo, Japan*.

Frédéric Béchet, Alexis Nasr, and Franck Genet. 2000. Tagging unknown proper names using decision trees. In *38th Annual Meeting of the Association for Computational Linguistics, Hong-Kong, China*, pages 77–84.

Yannick Estève, Christian Raymond, Renato De Mori, and David Janiszek. 2003. On the use of linguistic consistency in systems for human-computer dialogs. *IEEE Transactions on Speech and Audio Processing*, (Accepted for publication, in press).

Patrick Haffner, Gokhan Tur, and Jerry Wright. 2003. Optimizing SVMs for complex call classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'03*, Hong-Kong.

Y. He and S. Young. 2003. A data-driven spoken language understanding system. In *Automatic Speech Recognition and Understanding workshop - ASRU'03*, St. Thomas, US-Virgin Islands.

R. Kuhn and R. De Mori. 1995. The application of semantic classification trees to natural language understanding. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(449-460).

Mehryar Mohri, Fernando Pereira, and Michael Riley. 2002. Weighted finite-state transducers in speech recognition. *Computer, Speech and Language*, 16(1):69–88.

Christian Raymond, Yannick Estève, Frédéric Béchet, Renato De Mori, and Géraldine Damnati. 2003. Belief confirmation in spoken dialogue systems using confidence measures. In *Automatic Speech Recognition and Understanding workshop - ASRU'03*, St. Thomas, US-Virgin Islands.

Steve Young. 2002. Talking to machines (statistically speaking). In *International Conference on Spoken Language Processing, ICSLP'02*, pages 113–120, Denver, CO.