AUTOMATIC LEARNING OF INTERPRETATION STRATEGIES FOR SPOKEN DIALOGUE SYSTEMS

Christian Raymond[1], *Frédéric Béchet*[1], *Renato De Mori*[1], *Géraldine Damnati*[2], *Yannick Estève*[3]

[1] LIA/CNRS - University of Avignon, BP1228 84911 Avignon cedex 09 France
[2] France Télécom R&D - DIH/IPS/RVA 2 av. Pierre Marzin 22307 Lannion Cedex 07, France
[3] LIUM - Université du Maine, Avenue Laënnec, 72085 Le Mans Cedex 09
{*christian.raymond,frederic.bechet,renato.demori*}*@lia.univ-avignon.fr*
geraldine.damnati@rd.francetelecom.com yannick.esteve@lium.univ-lemans.fr

ABSTRACT

This paper proposes a new application of automatically trained decision trees to derive the interpretation of a spoken sentence. A new strategy for building structured cohorts of candidates is also described. By evaluating predicates related to the acoustic confidence of the words expressing a concept, the linguistic and semantic consistency of candidates in the cohort and the rank of a candidate within a cohort, the decision tree automatically learn a decision strategy for rescoring or rejecting a n-best list of candidates representing a user's utterance. A relative reduction of 18.6% in the Understanding Error Rate is obtained by our rescoring strategy with no utterance rejection and a relative reduction of 43.1% of the same error rate is achieve with a rejection rate of only 8% of the utterances.

1. INTRODUCTION

In recent years, decision trees have been proposed for dialogue control, dialogue act recognition, error detection in dialogue systems or for determining turns in dialogue [1, 2, 3].

This paper proposes a new application of automatically trained decision trees to derive the interpretation of a spoken sentence with a strategy that builds a cohort of candidates and then evaluates predicates about the acoustic confidence (AC) of the words expressing a concept, the linguistic (LC) and semantic (SC) consistency of candidates in the cohort and the rank (R) each candidate sentence would have by just considering the likelihood of its word sequence and the acoustic features.

With the same approach it is possible to find the probability that the interpretation is correct or that each of its component is correct. These results can be used by the dialogue strategy to decide if it should reason with the proposed interpretation or it should reject it by asking for a repetition, or if it is better to ask for a confirmation or a clarification about one or more property values.

The cohort of candidates is obtained by a network of Stochastic Finite State Transducers (SFST) introduced in [4] which can perform Language Model (LM) adaptation by boosting the probabilities of the transducers which provide at the output concept interpretations expected by the Dialogue belief. Rather than considering the N-best list of candidates, a *structured N-best list* is considered by merging the N-best lists (with N small) of word sequences for the most promising interpretations.

For each of the confidence parameters (AC, LC, SC and R) a mobile threshold to decide about acceptance or rejection is considered and the difference between the percentage of correctly and the wrongly accepted concepts is computed for each threshold value. Intervals are derived based on this difference function in order to adequately characterize situations where correct interpretation is predominant, situations in which errors are predominant and situations of uncertainty. These intervals are described by predicates appearing in questions used to train the decision strategy. Each leaf of the decision tree is associated with a probability that the interpretation carried by the candidate for which the leaf was reached is correct.

Based on the tree outcome, the Dialogue Manager can decide the action to take. A rescoring strategy is proposed in this paper that reorders the different candidates according to the probability assigned to them by the classification performed by the tree. Rejection can also take place based on these probabilities.

2. CONCEPT DECODING IN A SPOKEN DIALOGUE CONTEXT

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 concepts related to the application domain. Section 6 presents results obtained when system belief predicts the most frequent application dependent concepts, namely: *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

This research is supported by France Telecom's R&D under the contract 021B178

are: `Bastille` , `less+hundred+euros` and `indian`. In section 6, 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.

3. STRUCTURED N-BEST LISTS

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. An interpretation, for a given utterance, is simply the string of concepts that can be extracted from it. For example, by using the conceptual units presented in section 2, the following utterance transcription: *I'm looking for an Italian restaurant near Bastille around a hundred Euros* corresponds to the interpretation: `<FOOD_TYPE> <PLACE> <PRICE>`.

We propose here to structure the n-best word hypothesis list output by a Speech Recognition engine according to the various interpretations that can be found in every hypothesis of the list. The scores attached to each hypothesis remain the same. The interpretations are sorted according the score of their first word string hypothesis and the hypotheses among an interpretation are simply sorted according to their score.

A method for directly extracting such a structured n-best list from an utterance has been proposed in [4]. However, it is also possible, although at a higher computational cost, to first output a big number of hypotheses, then extract their interpretations (by means of regular grammars fro example) and finally build the structured n-best list.

4. CONFIDENCE MEASURES

4.1. 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.

4.1.1. Acoustic confidence measure at the concept level

For a word hypothesis *W* identified by the general model (λ_G) from frame t_0 to frame t_n, the likelihood of the corresponding speech signal *Y* is compared to the likelihood of the same portion of signal over an unconstrained phoneme loop. In order to be able to compare the values for different words, the estimated measure is actually the log-likelihood difference normalized by the number of frames over which it is computed.

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

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*.

In order to score the different concept hypothesis, the previous confidence measure easily extends to the concept level. In fact, 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 :

$$\Delta_{loop}(Y \mid \Gamma) = \frac{1}{\sum_{i=1}^{n} N_{frame}(W_i)} \times \sum_{i=1}^{n} N_{frame}(W_i)\Delta_{loop}(Y_i \mid W_i) \quad (2)$$

4.1.2. Acoustic confidence measure at the utterance level

As the decision tree framework described in the next section is able to handle confidence measures at different levels (word, concept and utterance), an acoustic measure is defined also at the utterance level. In a first approach the average Δ_{loop} over the whole utterance words is computed. When a solution in the n-best list is found to have a better average Δ_{loop} than the one-best solution, it can be assumed that the first-pass language model prevailed on the acoustic model for the one-best hypothesis generation.

4.2. 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 [5].

4.3. Semantic confidence measure (SC)

The two previous criterion give confidence measures at the word or at the concept unit level. However, in a Spoken Dialogue context, it's the *global interpretation* of an utterance which is relevant rather than its transcription. That's why we decided to add to our confidence measures a *semantic* criteria related to the global meaning of an utterance. This is done by means of a text classification approach. Several studies have shown that text classification tools (like Support Vector Machines or Boosting algorithms) can be an efficient way of labelling an utterance transcription with a semantic label such as a call-type [6] in a Spoken Dialogue context. In

our case, the semantic labels attached to an utterance are the concepts presented in section 2.

Three classifiers, one for each kind of concepts, have been trained on utterances extracted from our training and development corpora. To each utterance is attached 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. We also added to the training corpora the automatic transcriptions of the utterances in order to increase the robustness of the classifier to noisy data output by the ASR engine.

During the tagging process, the scores given by the text classifier are used as confidence scores. 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 [7].

4.4. Rank confidence measure (R)

To the previous confidence measures we added the rank of each candidate in its n-best. If it's a standard n-best list, the rank is simply the position in the list. If it's a structured n-best list, the 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. DECISION TREE BASED STRATEGY

5.1. From confidence measures to confidence labels

The first step in the specification of our decision tree based rescoring strategy is to define a training corpus. This corpus is made of automatic transcription of utterances from our test corpus. We chose to keep for each utterance all the n-best candidates (with n=12) contained in the standard or the structured n-best lists. To these transcriptions is also attached the concepts and the values detected by our SFST model as described in [4].

This corpus must also contains all the confidence criterion previously defined. The main advantage of a decision tree strategy is that one doesn't need to have any *a-priori* knowledge about the effectiveness of the criteria chosen: it's the decision tree itself that is going to select the relevant ones. However, because all the previous confidence scores are numerical values, one have first to find a discrete representation of their values. Two choices have to be done: how many discrete labels for representing the confidence values? and with which thresholds the conversion value→label is going to be done ?

On one hand, choosing a very small set of labels for representing a wide range of values leads to limit the discriminative power of the tree. On the other hand, having a large set of labels splits the training corpus in small sets of samples and may cause a data sparseness problem. We chose to have three different labels for the criterions AC, LC and SC: `H` for a high confidence, `N` for a neutral confidence and `L` for a low confidence. Because the size of the n-best lists to reorder is usually limited, we decided to represent each rank by a different label.

Therefore, to each sample of the decision tree training corpus are attached the following items: AC_{global}, the acoustic confidence label on the whole transcription; AC_H, the number of concepts detected and labelled with a high confidence; AC_N, the number of concepts detected and labelled with a neutral confidence; AC_L the number of concepts detected and labelled with a low confidence. Similarly we will have: LC_{global}, SC_H, SC_N, SC_L and the rank R.

As stated in the introduction, intervals of confidence measures are described by predicate labels. Intervals are obtained with a development corpus based on the difference between the percentage of correct acceptance *vs.* the percentage of false acceptance.

5.2. Rescoring process

In order to train the decision tree, we need first to give a label to each sample of the training corpus. Two labels are defined: `ALL_OK` and `NOT_OK`. The first one corresponds to samples whose concept/value items are all correct. The second one to samples containing at least one incorrect concept/value item.

The decision tree is then trained in order to minimize the impurity of the distribution of the `ALL_OK` and `NOT_OK` labels in the sets of samples. This process stopped when no further drop in impurity can be achieved or when the size of the set of samples attached to a node is below a given threshold. The questions used at each node of the tree are simply made from the confidence labels presented in section 5.1.

At the end of the training process, the score attached to each leaf of the tree is the ratio between the number of `ALL_OK` samples compared to the total number of samples in the set attached to the leaf. This score represents the confidence given by the classification process that a sample contains only correct concept/value items.

Once the tree is built, the rescoring process of a n-best list *L* is as follows:

- At first, all confidence scores for each candidate of *L* are calculated;
- the labels corresponding to the confidence scores are attached to each candidate;
- the tree is traversed by each candidate and the score attached to the leaf reached at the end of this process is given to the candidate;
- the candidate selected in *L* is the first one that received a score of being `ALL_OK` above a given threshold;
- finally if no candidate received a score above the threshold, the utterance is either rejected (strategy with rejection) or a back off candidate is chosen: the one with the highest score according to the tree (strategy with no rejection).

6. EVALUATION OF THE RESCORING STRATEGY

6.1. Experimental set-up

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 choose the different thresholds as presented in section 5.1 as well as training the rescoring decision tree. The Word Error Rate on the test corpus is 22.7%.

6.2. Evaluation of the rescoring strategy

Table 1 shows the results obtained with our rescoring strategy with no rejection on the development and test corpora. Two conditions are examined: rescoring a standard n-best list and rescoring a structured n-best list. The size of the n-best lists was set to 12 items: the first 12 candidates for the standard n-best list and the first 4 candidates of the first 3 interpretations for the structured n-best list. It is interesting to notice that the reduction in UER is significantly higher for structured n-best lists (from 14.4% to 21.5% on the development corpus and from 14.4% to 18.6% on the test corpus). This means that the decision tree take advantage of this structure to improve the split between the `ALL_OK` and the `NOT_OK` samples. In all cases, the gain obtained after rescoring with no rejection is very significant. 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.

Standard n-best lists			
Corpus	baseline	rescoring	UER reduction %
Devt.	12.1	10.4	14%
Test	16.7	14.3	14.4%
Structured n-best lists			
Corpus	baseline	rescoring	UER reduction %
Devt.	12.1	9.5	21.5%
Test	16.7	13.6	18.6%

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

Figure 1 shows the results obtained for the rescoring strategy with rejection on the structured n-best lists: when no candidate in a n-best list receives a score, by the decision tree, above a given threshold, the utterance is discarded. By changing this threshold we are able to plot a curve showing the Understanding Error Rate as a function of the utterance rejection rate. These results are presented according to the kind of criteria used to train the decision tree. As we can see, adding the rank of a candidate as a feature in the decision tree training is crucial. This can be explained by noticing that this rank is the only information the tree has about the score given by the ASR engine to a candidate. We can notice that the best results are obtained by using all the criteria available. A relative reduction of 43.1% of the error rate (from 16.7% to 9.5%) can be achieved with a rejection rate of only 8%.

7. CONCLUSION

This paper proposes a new application of automatically trained decision trees to derive the interpretation of a spoken sentence. A new strategy for building structured cohorts of candidates is also proposed. By evaluating predicates related to the acoustic confidence (AC) of the words expressing a concept, the linguistic (LC) and semantic (SC) consistency of candidates in the cohort and the rank (R) of a candidate within a cohort, the decision tree automatically learn a decision strategy for rescoring or rejecting a n-best

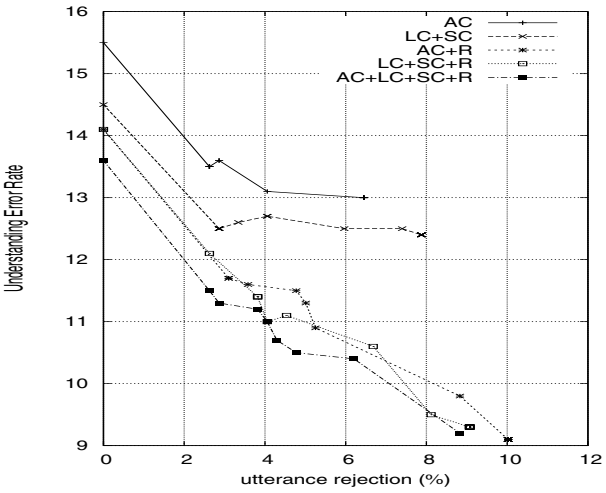

Fig. 1. Understanding Error Rate *vs.* utterance rejection: comparison of different criterion used to build the tree

list of candidates representing a user's utterance. A relative reduction of 18.6% in the Understanding Error Rate is obtained by our rescoring strategy with no utterance rejection and a relative reduction of 43.1% of the error rate is achieved with a rejection rate of only 8% of the utterances.

8. REFERENCES

[1] Kuansan Wang, "An event driven model for dialogue systems," in *International Conference on Spoken Language Processing, ICSLP'98*, Sidney, 1998, pp. 393–396.

[2] Rudnicky A. and Xu W., "An agenda-based dialog management architecture for spoken language systems," in *Automatic Speech Recognition and Understanding workshop - ASRU'99*, Keystone, CO, 1999.

[3] D. Stallard, "Flexible dialog management in the talk'n'train system," in *International Conference on Spoken Language Processing, ICSLP'02*, Denver, CO, 2002, pp. 2693–2696.

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

[5] Yannick Estève, Christian Raymond, Renato De Mori, and David 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] Patrick Haffner, Gokhan Tur, and Jerry Wright, "Optimizing SVMs for complex call classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'03*, Hong-Kong, 2003.

[7] R. Kuhn and R. De Mori, "The application of semantic classification trees to natural language understanding," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 449-460, 1995.