# Comparing ASR Modeling Methods for Spoken Dialogue Simulation and Optimal Strategy Learning

*Olivier Pietquin*[†]

Ecole Supérieure d'Electricité (Supélec)
Campus de Metz – STS Team
2 rue Edouard Belin, F-57070 Metz – France
olivier.pietquin@supelec.fr

*Richard Beaufort*

Multitel ASBL
Speech Processing Team
1 av. Copernic, B-7000 Mons – Belgique
richard.beaufort@multitel.be

## Abstract

Speech enabled interfaces are nowadays becoming ubiquitous. The most advanced ones rely on probabilistic pattern matching systems and especially on automatic speech recognition systems. Because of their statistical nature, performances of such systems never reach one hundred percent of correct recognition results. Performances are linked to environmental noise and to intra- and inter-speaker variability of course, but also to the acoustical similarities inside the vocabulary of allowed speech entries, which is usually contextual in the case of man-machine dialogue systems. A good dialogue strategy should therefore dynamically handle the potentiality of recognition errors. In this paper, we compare different methods to model ASR systems in the framework of automatic dialogue strategy optimization and we especially emphasize on a context-dependent ASR modeling method.

## 1. Introduction

Since more than a decade now, speech and language processing techniques have reached such a maturity level that speech enabled interfaces and Spoken Dialog Systems (SDS) are becoming ubiquitous. Strong skills in low level coding and signal processing are not required anymore to develop such systems thanks for example to dialog description languages such as VoiceXML [1]. Yet, the most challenging task in this context is to describe the optimal sequencing of speech and system interactions according to a balance between the speech and language processing performances and the usability of the system. Indeed, bad recognition results for example implies for the dialog strategy to include confirmation sub-dialogs while ergonomics rules would prevent a designer to systematically introduce confirmation sub-dialogs in the strategy.

Automatic Speech Recognition (ASR) has been a field of intensive researches for decades now. Today, the most successful systems are based on statistical methods such as Hidden Markov Models (HMMs)[2]. Because of this statistical nature, ASR systems can only achieve limited performances and will never reach maximal recognition rates (as well as humans). Performance bounds are influenced by external factors like environmental noise and/or intra- and inter-speaker variability but also by internal factors like acoustic models and acoustic similarities between word sequences accepted as speech entries, those allowed by the Language Model (LM). In most of SDSs, the LM (also called grammar) is generally context-dependent and allowed speech entries are different from one dialog state to another. A good dialogue strategy should therefore handle dynamically the potentiality of speech recognition errors, by introducing confirmation, help or recovery sub-dialogues or even avoid states in which a difficult speech recognition task is necessary.

Methods for automatic optimization of dialogue strategies have been proposed during the last decade [3][4][5]. All of them agree on the fact that artificial dialogue simulations are mandatory because of the huge amount of required data. Dialogue simulation implies, but not only, ASR modelling; that is simulating ASR errors and metrics generation (such as confidence level generation) without using real speech inputs. In this paper, we compare several ASR modelling methods in the framework of automatic dialogue strategy optimization. We especially compare the strategies learned when using different ASR models during dialogue simulations.

## 2. ASR Systems Features

The ASR process can be summarized as shown on Figure 1.

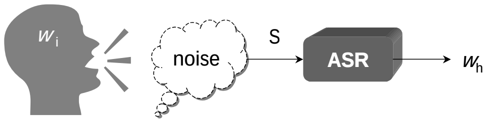

*Figure 1:* ASR Process

On this figure, a user utters a word sequence $\boldsymbol{w}_i = \{w_{i1}, w_{i2},\ldots,w_{in}\}$ in a noisy environment. The mixture of the speech signal and noise results in a new signal $\boldsymbol{S}$, which is processed by the ASR system. As an output (sometimes intermediate), the ASR system provides a new word sequence $\boldsymbol{w}_h = \{w_{h1}, w_{h2},\ldots,w_{hm}\}$ called the *hypothesis*. It is the result of a maximization process expressed by:

$$w_h = \arg\max_w P(w|S) = \arg\max_w \frac{P(S|w)\cdot P(w)}{P(S)} . \qquad (1)$$

ASR errors occur when $w_h \neq w_i$ that is when there has been a deletion (m < n), an insertion (m > n) or a substitution ($w_{ij}$

[†] This work was realized when the first author was with the Faculty of Engineering, Mons (Belgium) and was sponsored by the SIMILAR European Network of Excellence.

$\neq w_{hj}$). The World Error Rate (WER), which is a first ASR feature, is then defined as:

$$WER = \frac{Sub + Del + Ins}{|ASR \text{ tests}|} \times 100\,\% \ , \qquad (2)$$

where *Sub*, *Del* and *Ins* are namely the numbers of substitutions, deletions and insertions observed when performing |ASR tests| recognition tests with a given system.

Usually, an ASR system doesn't only provide a word sequence as a result but also associates a *confidence level* (CL) to this result. The CL is usually a real number between 0 and 1 (or an integer between 0 and 1000), based on acoustic measurements and defining how sure the system is about its recognition result [6]. The CL can be defined by 2 distributions (as shown on Figure 2) respectively for good and bad recognition results.

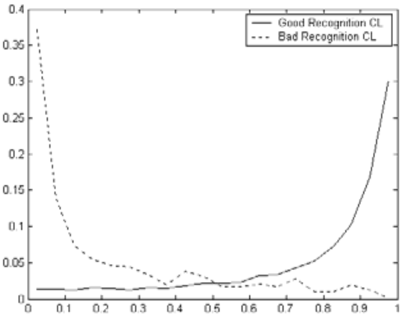

*Figure 2*: Confidence level distribution

An ASR model for dialogue simulation should therefore at least produce, without using real speech entries, word sequences showing errors according to a consistent WER and confidence levels drawn from consistent distributions.

## 3. Different ASR Models

In this section, we expose different ASR models that can be introduced in a modular, reusable and task independent simulation environment containing a user model and an ASR model like described in [7].

### 3.1. Simple ASR Model

As said in section 2, the ASR system can be modelled quite simply in we know the mean WER of the system and the CL distribution curves. Therefore, the first model tested in this paper produces errors according to a single WER and only one couple of CL distributions like those shown on Figure 2.

### 3.2. Task Classification

Eq. 1 is the general equation of ASR and it demonstrates that recognition confusions between two acoustically similar word sequences ($P(S|w_i) \approx P(S|w_j)$) can only happen when these word sequences are both allowed by the Language Model ($P(w) \neq 0$). Recognition performances should therefore be estimated according to the language model. Since the language model can change during a dialogue, it is responsible for context dependent ASR performance variations and the simple ASR model described in the previous section shouldn't lead to a high-quality dialogue strategy.

The first method we developed to take this dynamic LM dependency of ASR performances consist in classifying speech recognition tasks and assigning WER and CL distributions references to these tasks [7]. Speech recognition tasks can indeed be classified according to their difficulty and one can distinguish digits, numbers, dates, keywords, unrestricted continuous speech recognition,... This method requires for the dialogue system designer to assign a class to each LM that can be used during the interaction. This is not always possible since it happens that the LM is dynamically built during the interaction and its automatic classification is not trivial.

### 3.3. Automatic LM Adaptation

In order to obtain an automatic adaptation of the ASR model to the recognition task difficulty, we developed a method taking acoustical similarities between word sequences into account. The acoustical similarities are predicted from phonetic word transcription, so for each word sequence $w_j$ a set of phonetic transcription $\Phi(w_j)$ is supposed to be known:

$$\Phi(w_j) = \left\{\varphi^{\alpha}(w_j)\right\}_{\alpha=1}^{N_j} , \qquad (3)$$

where each $\varphi^{\alpha}(w_j)$ is a possible phonetic transcription itself composed of a sequence of phonemes (*symbols*) belonging to the alphabet $\Phi = \{\varphi_i\}$ of valid phonemes for the studied language:

$$\varphi^{\alpha}(w_j) = \left\{\varphi_k^{\alpha}(w_j)\right\}_{k=1}^{M_\alpha} . \qquad (4)$$

From this, an acoustic distance between two phonetic transcriptions can be drawn from the alignment cost between the associated phoneme sequences. The most popular method for measuring the difficulty of aligning two sequences is the *edit distance* [8], which can be efficiently computed by a Dynamic Programming (DP) algorithm. DP requires assessing costs to each edit operation and acoustic similarities should be introduced there. For deletions (insertions) there exists a single mapping between the symbol and the cost of deleting (inserting) the symbol $\varphi_i$ while for substitutions, a $|\Phi|$x$|\Phi|$ substitution cost matrix has to be built, where each element $c_{ij}$ is the cost of substituting symbol $\varphi_i$ to $\varphi_j$. To assess those costs the use of phonemes articulatory features is proposed. Indeed, a phoneme can be characterized by a type (vowel or consonant), an articulatory point (palatal, bilabial, …), lips positions and other features like the voiced/unvoiced nature. If $f_i$ ($f_j$) denotes the set of articulatory features of phoneme $\varphi_i$ ($\varphi_j$), a distance between both phonemes can be derived from a modified version of the Tverky's *ratio model*:

$$d(\varphi_i, \varphi_j) = \frac{F(f_i \setminus f_j)}{F(f_i \cap f_j) + F(f_i \setminus f_j) + F(f_j \setminus f_i)} . \qquad (5)$$

In (4), $F(.)$ is a function that applies on a set and assigns a weight to each element. One can notice that (4) defines an asymmetric distance. The $F(.)$ function can be derived from heuristics taught by linguistics:

- Confusions between vowels and consonants are unlikely.

- Confusions between consonants that only differ by the articulatory point are more likely.
- Confusions between vowels that only differ by the aperture degree are more likely and even more when augmenting the aperture (thus, the distance is asymmetric).
- A voiced consonant can become an unvoiced consonant when it is just before or after an unvoiced consonant.
- A vowel can be articulated with a different aperture degree when it is close to another vowel differing only by this feature ('close to' means in the surrounding syllables). It is even more likely when a tonic accent is on the other vowel.

The inter-phoneme distance given by (5) and computed thanks the previous considerations is used as a substitution cost in the DP algorithm. Linguistics also brings some clues about insertion and deletion costs:

- Deletions or insertions of liquid consonants are more likely than for other consonants.
- Deletions or insertions of the phoneme @ (*schwa*) are more likely than other vowels.
- Insertions or deletions of consonants are more likely immediately before or after another consonant.
- Deletions are more likely at the beginning and the end of a word and should then deserve a smaller cost.

With these parameters, the DP algorithm provides a distance between phoneme sequences that can be mapped to a confusion probability.

$$P\left(\varphi^{\alpha}(w_h) \mid \varphi^{\beta}(w_i)\right) \propto e^{-\lambda \cdot d\left(\varphi^{\alpha}(w_h), \varphi^{\beta}(w_i)\right)}. \tag{6}$$

Taking the LM into account and summing over all the possible phonetic transcriptions of each word sequence, one can derive the confusion probability between word sequences:

$$\sum_{\alpha=1}^{N_h} \sum_{\beta=1}^{N_h} P\left(\varphi^{\alpha}(w_h) \mid \varphi^{\beta}(w_i)\right) \cdot P_{LM}\left(\varphi^{\beta}(w_h) \mid \ell_h\right), \tag{7}$$

where $P_{LM}(\varphi^{\alpha}(w_h)|\ell_h)$ is the probability of occurance of the word sequence $w_h$ given the left context $\ell_h$ (the history) according to the LM. Using (7), we now have a method to transform a word sequence in another and simulate ASR errors. To test the validity of the method, we used (7) to compute the *acoustic perplexity* ($APP_{LM}$) [10] of the LM and compare it to the WER of a real ASR system. We actually used a modified version of $APP_{LM}$ so as to assume that there is no homophonous word in the set of the possible words at a given grammar state (in other words, the context can always help to distinguish homophonous words):

$$\overset{\wedge}{APP}{}_{LM}^{C,S} \approx \left( \prod_{w_h \in C} \frac{\sum_{\alpha=1}^{N_h} \sum_{\beta=1}^{N_h} P\left(s\left(\varphi^{\alpha}(w_h)\right) \mid \varphi^{\beta}(w_h)\right) \cdot P_{LM}\left(\varphi^{\beta}(w_h) \mid \ell_h\right)}{\sum_{w_i \in C} \sum_{\alpha=1}^{N_h} \sum_{\beta=1}^{N_i} P\left(s\left(\varphi^{\alpha}(w_h)\right) \mid \varphi^{\beta}(w_i)\right) \cdot P_{LM}\left(\varphi^{\beta}(w_i) \mid \ell_h\right)} \right)^{\frac{1}{|C|}}. \tag{8}$$

Results are shown on Figure 3. One can see that there is a quite good matching between the WER and $APP_{LM}$. The inter-word distance has also been used in this ASR model to simulate the production of CLs. The CL distributions have been approximated by a sum of two exponential curves:

$$P(CL \mid d) = \lambda_{low}(d) \cdot e^{\lambda_{low}(d) \cdot CL} + \lambda_{high}(d) \cdot e^{\lambda_{high}(d) \cdot CL}. \tag{9}$$

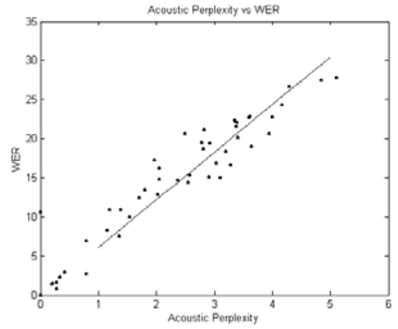

*Figure 3*: WER vs acoustic perplexity

In (9), $d$ is the mean inter-word distance in the considered LM. Figure 4 shows the predicted curve for good recognition and the histogram of real CL produced by an ASR system.

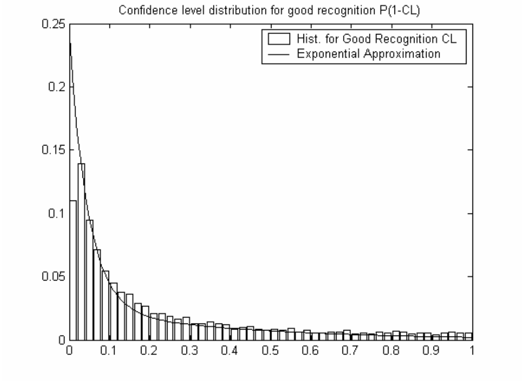

*Figure 4*: Prediction of CLs

## 4. Results

The three models described in section 3 were inserted in a simulation environment connected to a Reinforcement Learning (RL) agent [11] so as to optimize the dialogue strategy according to simulated performances. The dialogue task to optimize is the same as in [7] and consists in querying by speech a database including 350 computer configurations split into 2 tables (notebooks and desktops) containing 6 fields (or attributes) each: pc_mac (pc or mac), processor_type, processor_speed, ram_size, hdd_size and brand. The RL agent possible actions are greeting (greet), constraining questions (consQ), explicit confirmation (expC), confirmation of all attributes (allC), relaxation prompt (rel), database query (dbQ) and closing the dialog (clos). At each dialogue turn, an interaction cost $c_t$ is computed and serves as the reinforcement signal:

$$c_t = w_N \cdot N - w_{TC} \cdot TC - w_{CL} \cdot CL, \tag{10}$$

where $TC$ is a task completion measure, $CL_{ASR}$ is the ASR confidence level, $N$ is a measure of the dialog time duration (1 for each non-terminal turn and 0 otherwise) and $w_x$ are positive weights. The role of the RL agent is to find a strategy that minimise the mean cost of a complete dialogue. The dialogue states are built in an informational way that is they embed a summary of the information retrieved so far by the dialog system. Each state is then built on 7 Boolean values (one for each attribute) indicating whether its corresponding attribute has already been provided, a binary $CL_{ASR}$ value indicating if it is *high* or *low* for each attribute value, and a binary DB value indicating whether the last database query resulted in a *high* or *low* amount of retrieved data. Before each simulation, a user goal is defined as a database record and *TC* is defined as the mean number of common values between the retrieved records and the user's goal at the end of the dialog. The experiment is done with three different simulation environments, the first ($Sim_1$) is composed of a user model and the simple ASR, the second ($Sim_2$) contains the task-classification ASR model and the third ($Sim_3$) contains the adaptative ASR model.

The three experiments led to strategies in which after the greeting (supposed to be sufficiently open-ended to gather a lot of information at once), the agent either has enough information (with a *high* $CL_{ASR}$) to perform a DB query either it has to ask for more information from the user. If the DB query result set is empty, it has to prompt for the relaxation of some constrains. If the DB query result set is too large, the agent asks constraining questions. Before closing, it asks for a global confirmation. Table 1 shows the statistical properties of the learned strategies.

| | *N* | *TC* |
|---|---|---|
| $Sim_1$ | 7.99 | 6.1 |
| $Sim_2$ | 7.72 | 6.2 |
| $Sim_3$ | 7.65 | 6.3 |

| | greet | consQ | openQ | expC | allC | rel | dbQ | clos |
|---|---|---|---|---|---|---|---|---|
| $Sim_1$ | 1.00 | 1.57 | 1.24 | 0.33 | 1.32 | 0.31 | 1.22 | 1.00 |
| $Sim_2$ | 1.00 | 1.56 | 1.22 | 0.21 | 1.32 | 0.22 | 1.19 | 1.00 |
| $Sim_3$ | 1.00 | 1.55 | 1.22 | 0.20 | 1.33 | 0.24 | 1.22 | 1.00 |

*Table 1*: Results of strategy learning

The top three lines show the average number of turns *N* for each dialogue and the average task completion *TC* measured on 10,000 simulated dialogues. The following three lines are the average numbers of occurrence of each action in a simulated dialogue. One can see that going from $Sim_1$ to $Sim_3$ the number of turns decreases while the task completion increases. It also shows that the average number of explicit confirmations and relaxation prompts decrease, which means that the system follows a path where speech recognition results are getting better so it doesn't need to ask for confirmations.

A significant difference can be found in the order the constraining questions are asked as shown in Table 2.

| | note_ desk | pc_ mac | proc _ type | proc_ speed | ram size | hdd size | brand |
|---|---|---|---|---|---|---|---|
| $Sim_1$ | 0.1 | 0.1 | 0. 1 | 0.1 | 0.1 | 0.1 | 0.1 |
| $Sim_2$ | 0.2 | 0.2 | 0.01 | 0.1 | 0.1 | 0.1 | 0.01 |
| $Sim_3$ | 0.28 | 0.23 | 0.01 | 0.05 | 0.11 | 0.5 | 0.01 |

*Table 2*: Probability of each constQ action in initial state

With $Sim_1$ all the attributes have the same probability to be asked for. With $Sim_2$ (has described in [7]), to increase the overall confidence level and to reduce the average length of a dialog session (by avoiding useless confirmations) the agent asks questions about attributes that present better recognition results as numbers for example and prefers asking for a RAM size than a computer brand, yet the RAM size and the processor speed attributes have the same chance to be asked for because they are both classified as numbers. With $Sim_3$ the question order is changed because the system prefers asking for a RAM size than a processor speed because the possible values for the RAM size attribute in the DB where fewer and acoustically different. All attributes have different probabilities in this last experiment, which is a desirable behaviour.

## 5. Conclusions

In this paper we compared different ASR modeling methods in the framework of automatic optimization of spoken dialogue strategies. These ASR models are part of a modular dialogue simulation environment, they are independent from the task and can be interchanged. They reproduce the ASR behavior by generating error prone word sequences according to consistent word error rates and confidence levels according to consistent distributions. The results presented in section 4 show that they can be used to learn a strategy. This strategy can be modified according to predicted ASR performances in order to avoid as much as possible ASR problems either by introducing confirmation sub-dialogues or by avoiding problematic ASR tasks. This work extends the results obtained in [7] and it seems that the most sophisticated ASR model also produces a better strategy since the simulated dialogues are shorter and result in an improved task completion.

## 6. References

[1] S. McGlashan et al, W3C "Voice Extensible Markup Language (VoiceXML) Version 2.0", W3C Recommendation, http://www.w3.org/TR/voicexml20/, March 2004.

[2] L.R. Rabiner, "An Introduction to Hidden Markov Models", *in IEEE ASSP Magazine*, Jan. 1986, pp. 4-16.

[3] E. Levin, R. Pieraccini, W. Eckert, "Learning Dialogue Strategies within the Markov Decision Process Framework." *Proc. ASRU'97*, Santa Barbara, California, 1997.

[4] S. Singh, M. Kearns, D. Litman, M. Walker, "Reinforcement Learning for Spoken Dialogue Systems." *Proc. NIPS'99*, Denver, USA, 1999.

[5] O. Pietquin, *A Framework for Unsupervised Learning of Dialogue Strategies*. Presses Universitaires de Louvain, SIMILAR Collection, ISBN 2-930344-63-6, 2004.

[6] G. Williams, S. Renals, "Confidence Measures for Hybrid HMM/ANN Speech Recognition." *Proc. Eurospeech'97*, Rhodes, Greece, 1997, pp. 1955-1958.

[7] O. Pietquin, S. Renals, "ASR System Modeling for Automatic Evaluation and Optimization of Dialogue Systems." *Proc. ICASSP2002*, Orlando, USA, May 2002.

[8] V. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals." *Soviet Physics Daklady*, vol. 10, 1966, pp. 707-710.

[9] A. Tversky, "Features of Similarity." In *Psychological* Review, vol. 84, no. 4, 1977, pp. 327-352.

[10] H. Printz, P. Olsen, "Theory and Practice of Acoustic Confusability." *Proc. ISCA ITRW ASR2000*, Paris, France, 2000, pp. 77-84.

[11] R. Sutton, A. Barto, *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press, 1998.