ERROR SIMULATION FOR TRAINING STATISTICAL DIALOGUE SYSTEMS

Jost Schatzmann and Blaise Thomson and Steve Young

Cambridge University Engineering Department
Trumpington Street, Cambridge CB21PZ, UK
{js532,brmt2,sjy}@eng.cam.ac.uk

ABSTRACT

Human-machine dialogue is heavily influenced by speech recognition and understanding errors and it is hence desirable to train and test statistical dialogue system policies under realistic noise conditions. This paper presents a novel approach to error simulation based on statistical models for word-level utterance generation, ASR confusions, and confidence score generation. While the method explicitly models the context-dependent acoustic confusability of words and allows the system specific language model and semantic decoder to be incorporated, it is computationally inexpensive and thus potentially suitable for running thousands of training simulations. Experimental evaluation results with a POMDP-based dialogue system and the Hidden Agenda User Simulator indicate a close match between the statistical properties of real and synthetic errors.

Index Terms— error simulation, statistical modelling, spoken dialogue systems, POMDP, dialogue policy training

1. INTRODUCTION AND OVERVIEW

1.1. User and error simulation for training dialogue systems

A key advantage of taking a statistical approach to dialogue manager (DM) design is the ability to formalise design criteria as objective reward functions and to learn an optimal dialogue policy from real data [1]. The amount of suitably annotated in-domain data required for training a statistical system, however, typically exceeds the size of available dialogue corpora by several orders of magnitude. It is thus common practise to use a two-phased simulation-based approach where a statistical model of user behaviour is first trained on the limited amount of available human-computer dialogue data. The user simulator can then be used to generate any number of dialogues with the interactively learning DM (see [2, 3] for reviews).

Results presented by [4] indicate the benefit of training policies under noisy conditions. But while the development of user simulation tools is an active area of research, the error channel is often either excluded altogether [5, 6, 7] or simulated by generating random errors at a fixed error rate [8, 9, 10]. In real environments, errors typically depend on the acoustic confusability of the individual utterance and their frequency of occurrence is thus not a flat distribution over all words or semantic concepts. Errors are also highly system specific and depend not only on the speaker population and task specification but the system's speech recognition and understanding components as well as the type and amount of available training data. In the DARPA Communicator Project, for instance, sentence error rates ranged from 11.2 to 42.1% across the 9 participating sites [11]. A statistical approach to error modelling that allows the simulator to be trained on real errors produced with the actual spoken dialogue system (SDS) is therefore highly desirable.

1.2. Related literature on statistical error simulation

Previous work on statistical error simulation has investigated a number of different techniques. A straightforward approach is to condition the error rate on the type of task (eg. word vs. digit recognition) [12] and/or the individual speaker [13]. The simulated word error rate can also be set to approximate the distribution found in the training data: In [14, 15], for example, 70% of all utterances are transmitted with a WER of 0%, 10% with a WER of 100%, and 20% with a varying rate between 0 and 100%.

In [16] user dialogue behaviour is modelled as a network of interconnected states, with user actions corresponding to state transitions and errors corresponding to special "mumble"- or "null"-transitions. While the necessary statistics can be estimated on data, the recognition and understanding components are not explicitly modelled, and it is hence difficult to estimate the effects of individual system component improvements on the frequency of errors and the overall system performance.

All approaches described above avoid the poor assumption of a globally fixed error rate, but do not explicitly model the acoustic confusability of individual words and utterances. To overcome this limitation, error simulation based on phonetic confusions has been explored by a number of groups including [17, 18, 19, 20]. Word sequences are first mapped to phone sequences using a pronunciation dictionary and confusions are then generated using a set of probabilistic phoneme conversion rules [17], a handcrafted phone confusion matrix [18], or weighted finite state transducers [19, 20]. The corrupted sequence is then mapped back to a word sequence using the dictionary and (optionally) weighted using a language model.

While experiments with phone-level confusions have shown to produce promising results, the amount of training data needed to model context-dependent phone confusions for a typical tri-phone based recognizer is often very large. Generating a sufficient number of phonetic confusions in order to produce a list of word-level confusions containing semantically different utterances can also be computationally expensive and hence too slow to be useful for running thousands of dialogue simulations.

A computationally less expensive word-level error simulation method for training statistical dialogue systems has been suggested by [21]. Given a training corpus, a "Word Error Rate" is estimated for each word w_x by counting how many other words w_x is confused with. While providing an indicator of the confusability of each word, the method does not incorporate context-dependent features, and does not distinguish between substitution, deletion and insertion errors. Parameter estimation in [21] is carried out on an isolated word-recognition corpus, and there is no evaluation of whether or not the simulated error characteristics differ from those observed with real dialogue system recognition and understanding components.

2. A NOVEL APPROACH TO ERROR SIMULATION

2.1. Training SDS using Reinforcement-Learning

The application of statistical approaches to spoken dialogue systems, and in particular the use of Reinforcement-Learning techniques for optimal dialogue policy design has attracted significant interest over the last decade. The majority of work in this area is based on the well-known Markov-Decision Process (MDP) model, which serves as a formal representation of human-machine dialogue [5].

While MDPs provide a natural basis for modelling dialogue and have been widely studied in academia, their commercial impact has been minimal. This may be partly due to the fact that MDPs require the full state of the dialogue to be known exactly, and hence do not address the essence of the dialogue management problem, which is to handle the uncertainty present in human-computer dialogue arising from recognition and understanding errors [22, 23].

Partially-Observable MDPs (POMDPs) extend the MDP framework by maintaining a *belief space*, i.e. a probability distribution over multiple dialogue states. POMDPs thus incorporate an explicit model of uncertainty and present a much more powerful formalism for dialogue management. Their practical use in dialogue systems, however, is far from straightforward and it is only recently, that computationally tractable methods for modelling and updating the belief state and performing policy optimisation [22, 23, 24, 25] have been presented. Training experiments with an "agenda-based" user simulator [26, 7] have shown that competitive POMDP-policies can be learned, but as with MDP systems the required number of training episodes is high (typically $> 10^6$ dialogue turns). Computational efficiency is thus a key issue for error simulation techniques designed to train complex statistical systems.

2.2. Modelling human-computer dialogue errors

At a semantic level, human-computer dialogue can be viewed as a turn-based exchange of *dialogue acts*[1]. The dialogue act format used in this paper tags each user turn with an *act type* such as *hello*, *inform* or *request* and a list of zero or more *act items*. The complete dialogue act has the form *acttype(a=x,b=y,...)*, where the act items *a=x* and *b=y* denote slot-value pairs, such as *food=Chinese* or *pricerange=cheap*. To ensure a consistent probability distribution across multiple user act hypotheses in the POMDP model, every utterance is semantically decoded as a single dialogue act [27].

The error channel can be viewed as a generative probabilistic model $P(c, \tilde{a}_u|a_u)$, where a_u is the true incoming user dialogue act and $\tilde{a}_u$ is the recognised hypothesis with its associated confidence score c. (An extension to an n-best list of multiple hypotheses is possible.) For the purposes of error simulation, it is convenient to separate the confidence score generation from the error model, as has been previously suggested by [18, 22]

$$P(c, \tilde{a_u}|a_u) \quad = \quad P(c|\tilde{a_u}, a_u)P(\tilde{a_u}|a_u). \tag{1}$$

2.3. Capturing acoustic confusability

Maximum-Likelihood estimates for $P(\tilde{a_u}|a_u)$ can be easily obtained from an annotated corpus using frequency counting. In a typical corpus containing a few hundred dialogues, however, many user acts will never or rarely occur[2]. Back-off methods or parameter smoothing techniques can be applied, but finding a suitable scheme is difficult, because semantically "similar" dialogue acts are not necessarily acoustically similar. For example, while *inform(type=bar)* ("A bar please!") may be easily confused with *inform(drinks=beer)* ("Uh beer please!"), it is less likely to be confused with *inform(type=restaurant)* ("A restaurant please!").

It is thus desirable to model $P(\tilde{a_u}|a_u)$ in a way that allows word-level confusion statistics to be incorporated and estimated on real data. To achieve this, one may decompose the error model by summing over the joint probability of the recognised user act $\tilde{a}_u$, the recognised word sequence $\tilde{w}_u$ and the actual word sequence w_u, and then making reasonable conditional independence assumptions:

$$\begin{aligned} P(\tilde{a_u}|a_u) &= \sum_{\tilde{w_u}} \sum_{w_u} P(\tilde{a_u}, \tilde{w_u}, w_u|a_u) \\ &= \sum_{\tilde{w_u}} \underbrace{P(\tilde{a_u}|\tilde{w_u})}_{\substack{\text{semantic} \\ \text{decoder}}} \sum_{w_u} \underbrace{P(\tilde{w_u}|w_u)}_{\substack{\text{confusion} \\ \text{model}}} \underbrace{P(w_u|a_u)}_{\substack{\text{utterance} \\ \text{generation}}} \end{aligned} \tag{2}$$

The decomposed model shown in Eq. 2 consists of three components. $P(w_u|a_u)$ generates a word-level utterance for a given user act and is trained on user utterances seen in the dialogue corpus (Section 3). $P(\tilde{w_u}|w_u)$ simulates ASR confusions at the word-level and is trained using the reference transcriptions and ASR output recorded in the corpus (Section 4). $P(\tilde{a}_u|\tilde{w}_u)$ models the semantic decoding process and can be implemented by passing the generated utterance $\tilde{w}_u$ to the actual semantic decoder employed in the dialogue system. The confidence score generation process is described in Section 5.

3. UTTERANCE GENERATION MODEL

3.1. A maximum-likelihood approach

A generative Maximum-Likelihood model $P(w_u|a_u)$ for predicting a user utterance w_u for a given user act a_u is easily built by obtaining the appropriate relative frequency statistics from a transcribed and annotated dialogue corpus.

$$P(w_u|a_u) = \frac{f(w_u, a_u)}{f(a_u)} \tag{3}$$

During simulation, a word-level utterance w_u can then be generated for a_u according to the likelihood of w_u co-occurring with a_u in the training data. To resolve data sparsity issues and a possible lack of coverage for unseen dialogue acts, simple back-off templates can be created by replacing slot-values with general variables in seen utterances, as shown in the example below. Since the utterance generation model does not need to consider acoustic similarities, the templates can then be used to generate word-level utterances for semantically similar unseen user acts.

```
Source:    inform(food=Chinese, pricerange=cheap)
           CHINESE FOOD IN THE CHEAP PRICERANGE
Template:  inform(food=$X, pricerange=$Y)
           $X FOOD IN THE $Y PRICERANGE
Unseen:    inform(food=French, pricerange=expensive)
           FRENCH FOOD IN THE EXPENSIVE PRICERANGE
```

While more sophisticated methods of language generation exist (e.g. [28, 29, 30]), the technique presented here is computationally inexpensive and ensures complete coverage over the set of user acts.

[1] The terms dialogue act and action are used interchangeably in this paper.

[2] The dialogue act set used in this paper, for example, is roughly of the order of 10^3 and hence $P(\tilde{a_u}|a_u)$ can have up to 10^6 parameters.

4. ASR CONFUSION MODEL

4.1. Fragment-to-fragment alignments

At the word-level, ASR confusions can be viewed as translations of a source utterance w_u to a confused target utterance $\tilde{w}_u$. Omitting the subscript u for brevity, the source utterance w can be described as a sequence of S words, w_1^S, or a sequence of N fragments, f_1^N, where each fragment is a group of contiguous words in w.

Similarly, the target utterance $\tilde{w}$ may be viewed as a sequence of T words, $\tilde{w}_1^T$, or N confused fragments, $\tilde{f}_1^N$. Note that while the length S of the source utterance does not necessarily equal the length T of the target utterance, it can be assumed that the number N of "clean" source fragments matches the number of "confused" target fragments. This assumption can be made without loss of generality since each fragment can have 0 or more words (cf. Fig. 1).

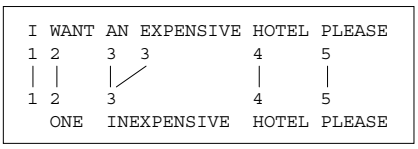


Fig. 1. A sample source and target alignment. The central modelling assumption is that each fragment $\tilde{f}_i$ in the confused utterance is generated as a "translation" of the fragment f_i in the source utterance: $f_i \rightarrow \tilde{f}_i$. Since the length of a fragment can vary, this allows substitution, insertion and deletion errors to be modelled. If $f_i = \tilde{f}_i$, this corresponds to the correct recognition of the given fragment.

In order to formally define the word-to-fragment alignment, it is useful to introduce the mapping functions γ and $\tilde{\gamma}$ for mapping word indices to fragment indices. Using a vector style notation, one may write $\gamma = \gamma_1^S$, with $\gamma_k = i$ iff the k'th word in the source utterance w belongs to the source fragment f_i. Similarly, $\tilde{\gamma} = \tilde{\gamma}_1^T$ governs the alignment of the confused target utterance $\tilde{w}$, with $\tilde{\gamma}_k = i$ iff the k'th word in $\tilde{w}$ belongs to fragment $\tilde{f}_i$.

Letting n_i denote the length of the clean fragment f_i, and $\tilde{n}_i$ denote the length of the confused fragment $\tilde{f}_i$, the two constraints $\sum_{i=1}^N n_i = S$ and $\sum_{i=1}^N \tilde{n}_i = T$ ensure that the combined length of the clean (confused) fragments matches the length of the source (target) word sequence. Our modelling objective can now be described as finding the conditional distribution

$$P(\tilde{w}, \tilde{\gamma}, \gamma | w) = P(\tilde{w}, \tilde{\gamma} | \gamma, w) P(\gamma | w). \quad (4)$$

4.2. A simple alignment model

Assuming that the alignment of each word depends only on the current word and the previous fragment, the probability of a source alignment γ for a given word sequence w can be expressed as

$$P(\gamma|w) = P(\gamma_1^S | w_1^S) = P(\gamma_1|w_1) \prod_{i=2}^{S} P(\gamma_i | w_i, w_{start}^{i-1}, \gamma_{i-1}) \quad (5)$$

where w_{start}^{i-1} are the words assigned to γ_{i-1}. The word w_1 is necessarily assigned to fragment f_1, hence $P(\gamma_1|w_1) = 1$ iff $\gamma_1 = 1$ and 0 otherwise. For all subsequent words, the probability of assigning w_i to fragment γ_i given that the words w_{start}^{i-1} have been assigned to γ_{i-1} can be defined as

$$P(\gamma_i | w_i, w_{start}^{i-1}, \gamma_{i-1}) = \begin{cases} \phi & \text{if } \gamma_i = \gamma_{i-1} \\ 1 - \phi & \text{if } \gamma_i = \gamma_{i-1} + 1 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where ϕ is the likelihood of seeing w_i follow w_{i-1} in the fragment starting with w_{start}

$$\phi = \frac{freq(w_{start} \ldots w_{i-1} w_i)}{freq(w_{start} \ldots w_{i-1})}. \quad (7)$$

When simulating errors, the alignment model is used in a generative fashion. It is hence not necessary to compute the probability of all possible word-to-fragment alignments for w. Instead, only one alignment needs to be generated, requiring a single pass through the given word sequence.

4.3. Fragment confusions

Once the alignment γ of the source w is known, the conditional probability of the target $\tilde{w}$ and its alignment $\tilde{\gamma}$ given γ and w can be modelled using fragment confusion probabilities

$$P(\tilde{w}, \tilde{\gamma} | \gamma, w) = P(\tilde{f}_1^N | f_1^N) = \prod_{i=1}^{N} P(\tilde{f}_i | f_i, \tilde{f}_1^{i-1}) \quad (8)$$

$$\approx P(\tilde{f}_1^N) \prod_{i=1}^{N} P(\tilde{f}_i | f_i). \quad (9)$$

Note that the conditioning of each target fragment $\tilde{f}_i$ in Eq. 8 is only on the corresponding source fragment f_i and the preceding target fragments. The approximation shown in Eq. 9 respects these conditional independence assumptions and can be implemented in a computationally tractable manner using an over generate-and-sample approach: First, an n-best list of hypotheses is generated by applying the confusion model $P(\tilde{f}_i|f_i)$ multiple times to the clean source fragment sequence. Exploiting the fact that $P(\tilde{f}_1^N) = P(\tilde{w})$, all n hypotheses are then scored using the dialogue system's language model. The ASR output is selected by sampling from the list of hypotheses according to their language model probabilities.

To obtain the necessary fragment confusions statistics, all pairs of reference transcriptions and ASR outputs $(w, \tilde{w})$ in the training corpus are aligned using a Levenshtein distance matrix such that the cost of transforming w into $\tilde{w}$ is minimal given a fixed cost for inserting, deleting and substituting words. The result is a lookup-table of all fragments occurring in the training transcriptions, together with their possible confusions (see sample entry below), and the statistics needed for Eqs. 7 and 9 can be easily obtained from it.

```
"A BAR" -> "ALL","ART","A BAR","A BAR",
           "A CAR","BAR","BAR","BAR","BAR","CAR";
```

5. CONFIDENCE SCORE GENERATION

Speech recognition engines for dialogue systems typically associate a confidence score c with each recognition hypothesis to indicate the reliability of the result. As has been previously demonstrated by [18, 22], it is convenient to approximate $P(c|\tilde{a}_u, a_u)$ by assuming that there are two distributions for c, one if $\tilde{a}_u$ matches a_u and one if it does not.

$$P(c|\tilde{a_u}, a_u) \approx \begin{cases} P_{corr}(c) & \text{if } \tilde{a_u} = a_u \\ P_{incorr}(c) & \text{if } \tilde{a_u} \neq a_u \end{cases} \quad (10)$$

In [18, 22], the two distributions are handcrafted. For the experiments presented in this paper, confidence scores for correct and incorrect hypotheses are generated by sampling from the distributions found in the training data.

6. EVALUATION

6.1. Model training and evaluation setup

The statistical models presented in this paper are trained on a transcribed and annotated corpus of human-computer dialogues recorded with the POMDP-based Hidden Information State (HIS) Dialogue System [23, 25]. The HIS system is a Tourist-Information domain prototype that helps users find hotels, bars, and restaurants in a fictitious town, subject to certain constraints. (E.g. "a cheap Chinese restaurant near the Post Office" or "a wine bar playing Jazz music on the Riverside"). The dataset consists of 160 dialogues, recorded with 40 different speakers (each of whom completed 4 dialogues) and contains a total of 6452 dialogue turns and 21,667 words.

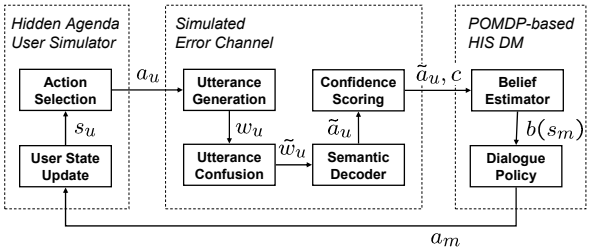


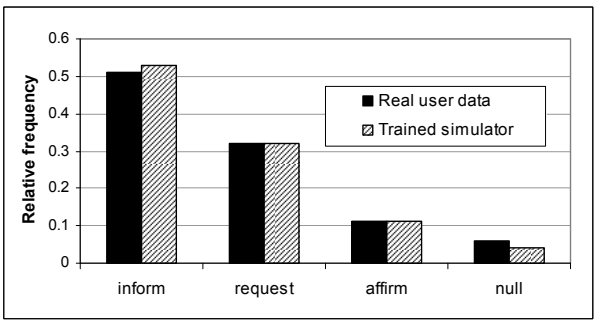
Fig. 2. Dialogue simulation framework

To evaluate the simulated error channel, a corpus of 10,000 synthetic dialogues is generated using the setup illustrated in Fig. 2. System dialogue acts are produced with the same HIS Dialogue Manager used to record the corpus described above and user dialogue acts are generated using the Hidden Agenda User Simulator [7]. In each dialogue cycle, the user output a_u is first translated to a word-level utterance w_u and corrupted with ASR confusions to form a list of hypotheses. The ASR result $\tilde{w}_u$ is obtained by sampling from this list using the HIS system language model, as described in Section 4 and parsed by the HIS semantic decoder to form $\tilde{a}_u$. The result is associated with a synthetic confidence score and passed to the HIS dialogue manager to generate the machine response a_m, which in turn is fed back to the user simulator.

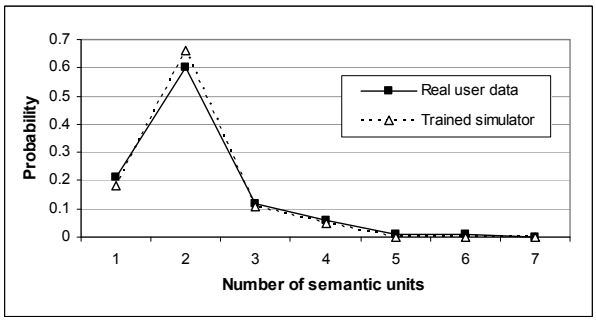
6.2. Evaluation of generated user utterances

Before evaluating the word-level utterance generation model, it is useful to assert that the semantic-level output of the user simulator correctly reproduces the statistical properties of real user dialogue acts. Fig. 3 (a) shows the relative frequency of the 4 most common user dialogue acts in real and simulated data and Fig. 3 (b) shows the distribution over the number of *semantic units* per dialogue act. It is assumed that the dialogue act type and each dialogue act item (slot-value pair) count as one semantic unit. The act *affirm(area=north,stars=2)*, for instance, contains 3 semantic units. As shown, the distribution over semantic units and dialogue act types selected by the simulated user is very similar to that of real users.

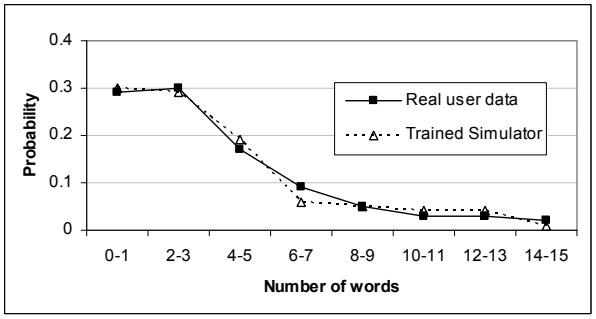
The utterance generation model can now be evaluated by comparing statistical properties of the simulated utterances with those of real user utterances. Fig. 3 (c) shows the distribution over the number of words per utterance and illustrates that the length of the synthetic utterances is similarly distributed as the length of real utterances. In both cases, more than half of all utterances are very short (less than 4 words). The correlation between utterance length and the number of semantic units is shown in Fig. 3 (d): Here all utterances are grouped into bins according to their number of words. The mean number of semantic units per bin is then plotted and again it can be seen that the simulated data has similar properties as the real data.



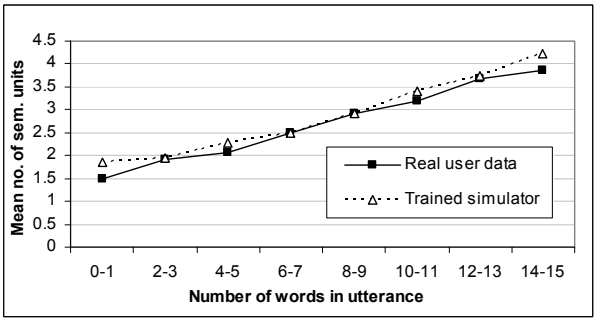
(a) Distribution over dialogue act types



(b) Distribution over number of semantic units per utterance

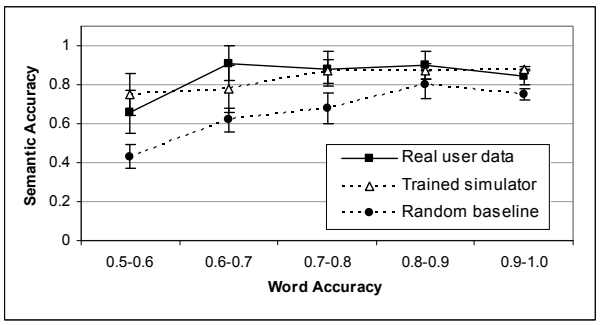


(c) Distribution over number of words per utterance

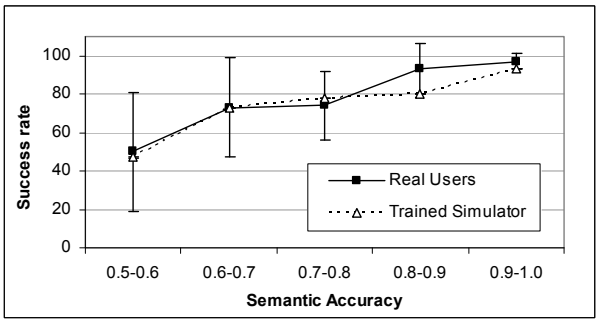


(d) Utterance length vs. semantic units

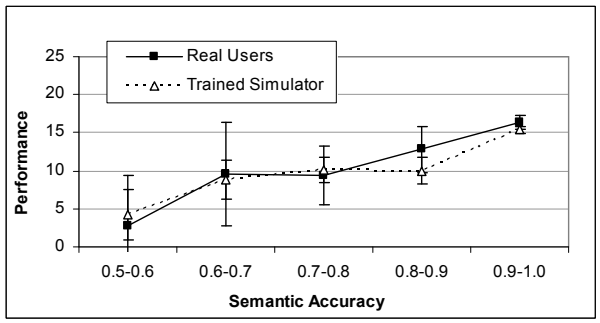
Fig. 3. Comparison of real and simulated utterances



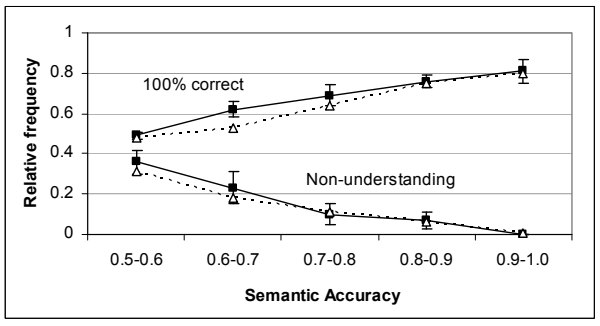
(a) Word Accuracy vs Semantic Accuracy



(b) Semantic Accuracy vs Success Rate



(c) Semantic Accuracy vs Dialogue Performance



(d) Semantic Accuracy vs Error-free and Non-understanding

Fig. 4. Comparison of real and simulated ASR errors

6.3. Evaluation of synthetic ASR confusions

An effective method for evaluating the simulated ASR confusions is to explore the correlation between Word Accuracy (WAcc) and Semantic Accuracy (SAcc). Based on the number of substitutions, insertions and deletions in an utterance containing W words and SU semantic units, one may define WAcc [31] and SAcc [32] as

$$\text{Semantic Accuracy} = 1 - \frac{SU_S + SU_I + SU_D}{SU} \quad (11)$$

$$\text{Word Accuracy} = 1 - \frac{W_S + W_I + W_D}{W}. \quad (12)$$

For Fig. 4 (a), all user utterances are grouped into bins according to their WAcc and the mean SAcc is then computed for each bin. As shown in the figure, a decrease in WAcc down to around 60% in real data does not lead to a significant drop in SAcc. This may be explained by the fact that the word confusions in this range often affect non-concept-words so that the decoder is still able to recognize most semantic units correctly. While the trained simulator mirrors this phenomenon reasonably well, a baseline using random word confusions produces a much larger drop in SAcc, since all words are equally likely to be confused.

For Figs. 4 (b-d), all dialogues were grouped in bins according to their average SAcc. In (b), the success rate is then computed for each bin, i.e. the percentage of dialogues where a correct venue was recommended. For (c), the average dialogue performance per bin is computed by assigning 20 points for a successful venue recommendation (0 otherwise) and subtracting a 1 point penalty for every dialogue turn. Fig. 4 (d) shows the relative frequency of utterances which are 100% semantically correct and the relative frequency of utterances that can be classified as non-understanding errors (0% semantically correct). All figures indicate a close match between the statistical properties of the real and simulated data, and in many cases no statistically significant difference is found.

6.4. Evaluation of generated confidence scores

As explained in Section 5, confidence scores are generated by sampling from the distribution of confidence scores seen in the training data. The distribution over simulated confidence scores hence closely matches the real distribution, as verified by Fig. 5.

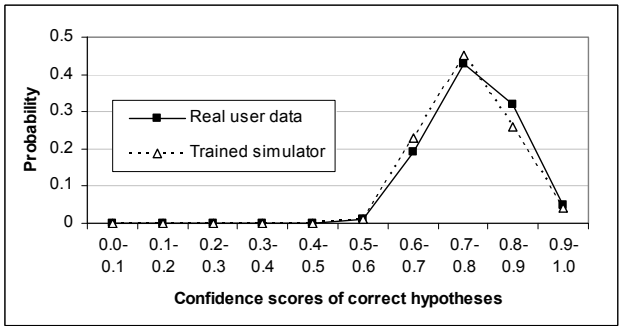


Fig. 5. Graph showing the prob. of the confidence score falling into one of the ten confidence score intervals, given that the hypothesis is correct. A similar graph may be plotted for incorrect hypotheses.

7. SUMMARY

This paper has presented a novel and computationally inexpensive approach to error simulation, suitable for generating large numbers of training episodes for statistical dialogue systems. Based on word-level utterance generation and ASR confusion models, it explicitly models the context-dependent acoustic confusability of words and allows the system specific language model and semantic decoder to be incorporated. Confidence scores are obtained by sampling from the distribution observed in the training data. Experimental results show that the models can be successfully trained on a small corpus of transcribed and annotated data and that the statistical properties of the simulated utterances and dialogue errors closely match those observed in real human-computer dialogue data.

8. REFERENCES

[1] S. Young, "Talking to machines (statistically speaking)," in *Proc. ICSLP*, 2002.

[2] J. Schatzmann, K. Weilhammer, M.N. Stuttle, and S. Young, "A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies," *KER*, vol. 21, no. 2, pp. 97–126, 2006.

[3] O. Lemon and O. Pietquin, "Machine learning for spoken dialogue systems," in *Proc. Eurospeech, Antwerp, Belgium*, 2007.

[4] O. Lemon and X. Liu, "Dialogue policy learning for combinations of noise and user simulation: transfer results," in *Proc. SIGDial, Antwerp, Belgium*, 2007.

[5] E. Levin, R. Pieraccini, and W. Eckert, "A Stochastic Model of Human-Machine Interaction for Learning Dialog Strategies," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 1, pp. 11–23, 2000.

[6] V. Rieser and O. Lemon, "Cluster-based User Simulations for Learning Dialogue Strategies.," in *Proc. ICSLP*, 2006.

[7] J. Schatzmann, B. Thomson, and S. Young, "Statistical User Simulation with a Hidden Agenda," in *Proc. SIGDial, Antwerp, Belgium*, 2007.

[8] K. Hone and C. Baber, "Using a simulation method to predict the transaction time effects of applying alternative levels of constraint to user utterances with speech interactive dialogs," in *ESCA Workshop on SDS*, 1995, Vigso, Denmark.

[9] M. Araki, T. Watanabe, and S. Doshita, "Evaluating dialogue strategies for recovering from misunderstandings," in *In Proc. IJCAI Workshop on Collaboration Cooperation and Conflict in Dialogue Systems*, 1997, pp. 13–18.

[10] T. Watanabe, M. Araki, and S. Doshita, "Evaluating dialogue strategies under communication errors using computer-to-computer simulation," *Trans. of IEICE, Info Syst.*, vol. E81-D, no. 9, pp. 1025–1033, 1998.

[11] M. Walker et al., "DARPA Communicator: Cross-system results for the 2001 evaluation," in *Proc. ICSLP*, 2002.

[12] O. Pietquin and S. Renals, "ASR System modeling for Automatic Evaluation and optimization of of Dialogue Systems," in *Proc. ICASSP, Orlando, FL, USA*, 2002.

[13] T. Prommer, H. Holzapfel, and A. Waibel, "Rapid Simulation-Driven Reinforcement Learning of Multimodal Dialog Strategies in Human-Robot Interaction," in *Proc. ICSLP*, 2006.

[14] K. Georgila, J. Henderson, and O. Lemon, "Learning user simulations for information state update dialog systems," in *Proc. Eurospeech, Lisbon, Portugal*, 2005.

[15] O. Lemon, K. Georgila, and J. Henderson, "Evaluating Effectiveness and Portability of Reinforcement Learned Dialogue Strategies with real users: the TALK TownInfo Eval," in *Proc. SLT, Palm Beach, Aruba*, 2006.

[16] K. Scheffler and S. Young, "Corpus-based dialogue simulation for automatic strategy learning and evaluation," in *Proc. NAACL Workshop on Adaptation in Dialogue Systems*, 2001.

[17] Y. Deng, M. Mahajan, and A. Acero, "Estimating speech recognition error rate without acoustic test data," in *Proc. Eurospeech, Geneva, Switzerland*, 2003.

[18] O. Pietquin, *A Framework for Unsupervised Learning of Dialogue Strategies*, Ph.D. thesis, Polytech de Mons, 2004.

[19] E. Fosler-Lussier, I. Amdal, and H.-K. J. Kuo, "On the road to improved lexical confusability metrics," in *Proc. of PMLA*, Estes Park, Colorado, 2002.

[20] M. Stuttle, J. Williams, and S. Young, "A framework for dialog systems data collection using a simulated asr channel," in *Proc. ICSLP*, 2004.

[21] O. Pietquin and T. Dutoit, "A probabilistic framework for dialog simulation and optimal strategy learning," *IEEE Trans. on Speech and Audio Processing, Special Issue on Data Mining of Speech, Audio and Dialog*, 2005.

[22] J. Williams and S. Young, "Partially Observable Markov Decision Processes for Spoken Dialog Systems," *Computer Speech and Language*, vol. 21, no. 2, pp. 231–422, 2007.

[23] S. Young, "Using POMDPs for Dialog Management," in *Proc. SLT, Palm Beach, Aruba*, 2006.

[24] S. Young, J. Schatzmann, K. Weilhammer, and H. Ye, "The Hidden Information State Approach to Dialog Management," in *Proc. ICASSP, Honululu, HI, USA*, 2007.

[25] B. Thomson, J. Schatzmann, K. Weilhammer, H. Ye, , and S. Young, "Training a real-world POMDP dialogue system," in *Proc. of HLT/NAACL Workshop: Bridging the Gap*, 2007.

[26] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young, "Agenda-based User Simulation for Bootstrapping a POMDP Dialogue System," in *Proc. of HLT/NAACL*, 2007.

[27] S. Young, J. Williams, J. Schatzmann, M. Stuttle, and K. Weilhammer, "The Hidden Information State Approach to Dialogue Management," Tech. Rep. CUED/F-INFENG/TR.544, Cambridge University, 2005.

[28] I. Langkilde and K. Knight, "Generation that exploits corpus-based statistical knowledge," in *Proc. ACL*, 1998.

[29] S. Seneff, "Response planning and generation in the mercury flight reservation system," *Computer Speech and Language*, vol. 16, pp. 283–312, 2002.

[30] C. Wang, S. Seneff, and G. Chung, "Language model data filtering via user simulation and dialogue resynthesis," in *Proc. Eurospeech, Lisbon, Portugal*, 2005.

[31] L. Hirschman and H. Thompson, "Overview of evaluation in speech and natural language processing," in *In R. Cole, editor, Survey of the State of the Art in Human Language Technology. Cambridge University Press*, 1996.

[32] M. Boros, W. Eckert, F. Gallwitz, G. Gorz, G. Hanrieder, and H. Niemann, "Towards understanding spontaneous speech: Word accuracy vs. concept accuracy," in *Proc. ICSLP*, 1996.