Designing a spoken language interface for a tutorial dialogue system

Peter Bell, Myroslava Dzikovska, Amy Isard

School of Informatics, University of Edinburgh, UK
{peter.bell,m.dzikovska,amy.isard}@ed.ac.uk

Abstract

We describe our work in building a spoken language interface for a tutorial dialogue system. Our goal is to allow natural, unrestricted student interaction with the computer tutor, which has been shown to improve the student's learning gain, but presents challenges for speech recognition and spoken language understanding. We discuss the choice of system components and present the results of development experiments in both acoustic and language modelling for speech recognition in this domain.

Index Terms: spoken dialogue system, speech recognition, computer tutoring, adaptation

1. Introduction

Most research in spoken dialogue systems has focussed on systems which are task-oriented, designed to help the user achieve some fixed goal in a minimum number of dialogue turns, often using a slot-filling paradigm. We believe that spoken dialogue systems could be deployed more widely in the domain of computer tutoring, where, in contrast, the primary aim is to maximise the student's learning gain from using the system.

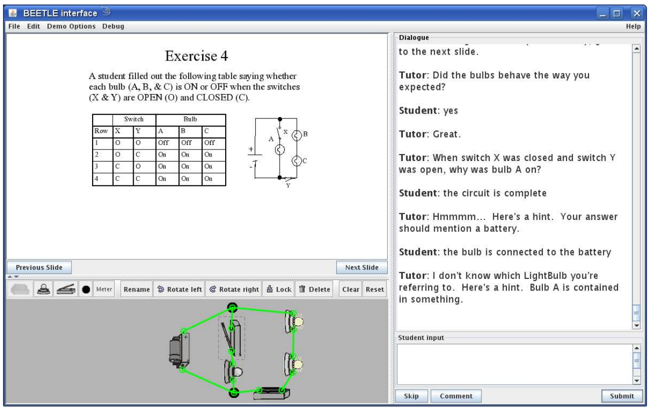
A substantial body of research eg. [1, 2] has shown that an effective tutoring technique is to encourage students to produce their own explanations and generally to talk more about the domain during problem-solving. This motivated the development of dialogue-based intelligent tutoring systems (ITS) which ask students open-response questions (rather than multiple-choice questions), and in particular explanation questions. However, to date such systems have largely been limited to using typed interactions; existing speech-enabled tutorial dialogue systems such as [3, 4] have been constrained to small-vocabulary scenarios which restrict the student to a limited range of answers, and therefore restrict opportunities for self-explanation.

Recently, typed systems attempted to encourage long, open-ended student answers through asking explanation questions and giving targeted feedback on explanations. While this approach was effective in some cases [5], there is evidence that in human-human tutoring spoken dialogue is more effective than typing [6]. There is also evidence that self-explanation can be encouraged by the addition of a spoken language interface [7], and this is also borne out by our experience during pilot data collection, an example of which is shown below in Figure 4.

Therefore, we are investigating the feasibility of adding the capability for natural, unrestricted spoken interaction to BEETLE II, our existing typed tutorial dialogue system which supports open-ended explanation questions [8]. The BEETLE II system teaches students basic electricity and electronics by introducing them to the fundamental concepts, then challenging their understanding by asking them to predict the outcome of exercises in a circuit simulator and to explain their reasoning. This approach encourages deep thinking, requiring the students to articulate the causes and effects of the events they are observing. The version of the BEETLE II tutor deployed currently uses typed interaction; it is capable of analysing complex student utterances and automatically generating feedback using a hand-coded feedback policy based on analysis of a corpus of human-human tutoring dialogues. The example dialogue shown in Figure 1 illustrates the natural language capabilities of the system; a screenshot of the Graphical User Interface (GUI) is shown in Figure 2.

Tutor:	Why was bulb A on when switch Y was open and switch Z was closed?
Student:	because it had a closed path
Tutor:	Right. There is a closed path. Keep going.
Student:	it has a closed path with the battery
Tutor:	You're on the right track. Here's the answer. Bulb A is still contained in a closed path with the battery.

Figure 1: *Example interaction with the system from the* BEETLE II *corpus*



Exercise 4

A student filled out the following table saying whether each bulb (A, B, & C) is ON or OFF when the switches (X & Y) are OPEN (O) and CLOSED (C).

	Switch		Bulb		
Row	X	Y	A	B	C
1	O	O	Off	Off	Off
2	O	C	On	On	On
3	C	O	On	On	On
4	C	C	On	On	On

to the next slide.

Tutor: Did the bulbs behave the way you expected?

Student: yes

Tutor: Great.

Tutor: When switch X was closed and switch Y was open, why was bulb A on?

Student: the circuit is complete

Tutor: Hmmmm... Here's a hint. Your answer should mention a battery.

Student: the bulb is connected to the battery

Tutor: I don't know which LightBulb you're referring to. Here's a hint. Bulb A is contained in something.

Figure 2: *Screenshot of the current*BEETLE II *tutor with text-based interaction*

To our knowledge, the addition of speech modality to complement the NLP capabilities of BEETLE II will make it the first ITS capable of processing long spoken explanation answers. Moving from typed to spoken interactions in this type of system presents a number of challenges. This paper provides an overview of our work in overcoming these, focussing primarily on the construction of the capability for automatic speech recognition (ASR) in this domain. We give a brief overview of the system components in Section 2, describe our approach to language modelling in Section 3, and discuss acoustic modelling in Section 4, giving results on development data within each section. Section 5 considers future work.

2. Architecture

The system is highly modular in design, illustrated in figure 3. On the input side, the system employs a deep parser, TRIPS [9] which provides a domain-independent semantic representation, followed by higher-level domain reasoning and diagnostics components which determine the correctness of student explanations. Based on this input, the tutorial planner module selects which tutorial strategy to use, which is implemented via a deep generation module which constructs tutorial feedback using a domain-specific content planner together with relevant content from the student's own answer.

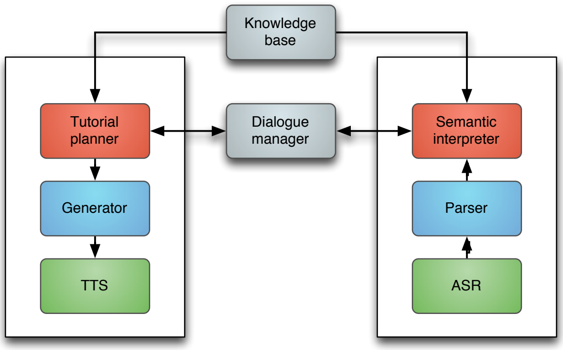


Figure 3: *The modules of the* BEETLE II *system*

The new ASR module uses ATK[1] to perform one-line speech parametrisation, voice activity detection and speech recognition in real-time using a multi-threaded design (though dialogue-management functions are delegated to the existing BEETLE II dialogue manager). We replaced ATK's native Viterbi decoder with our own online version of HTK's HDecode, to allow efficient large-vocabulary recognition. Online speaker adaptation is performed. The language modelling and acoustic modelling design choices made are described in more detail below. Spoken output is provided using the Festival text-to-speech engine. Whilst the ASR operates in an online mode, all speech is recorded for later analysis. In addition to the natural language components, the GUI includes an area to display reading material and an interactive circuit simulator.

3. Language modelling

In many spoken dialogue systems, ASR is performed using hand-crafted finite-state networks selected according to the dialogue state. This is not appropriate for our system, where it is important to allow unrestricted speech, at least in principle, because students often struggle with unfamiliar terminology: effective tutoring requires knowing the words that the student said, even if they are out of domain. Therefore recognition is performed using an n-gram language model (LM).

We have a corpus available of domain-specific data comprising 90,000 words of typed interactions with the earlier BEETLE II system, collected during 2009, which we denote `student09`. However, we would expect the lexical content of the spoken input to differ considerably from to the typed inputs: the switch to the spoken modality is likely to result in more verbose responses, and furthermore, the speech may contain disfluencies characteristic of spontaneous speech – filled pauses, repetitions, deletions and repairs – particularly to be expected since the users are presented with often challenging, open-ended questions which may cause hesitation and uncertainty. As an illustration of this, Figure 4 shows an example of two different spoken student responses from our development data, illustrating the contrast with typed answers.

Student one: Row one. If bulb A is out bulb B and C will remain on. So number one is correct. Row two. Bulb B is out therefore bulb C will be out so that is incorrect and vice versa for row number three. If C is out B will also be out.

Student two: X is it open? Row two is incorrect. Um. Row three is incorrect. Rows two and three are incorrect.

Figure 4: *Two example responses to the question "Which rows do you think are incorrect?" from our development collection of spoken interaction. To make the text more readable here, punctuation has been added based on features of the recordings.*

To solve this problem, we used two further corpora. Firstly the Fisher corpus, comprising around 1,000,000 words from transcribed telephone conversations, was used to obtain natural speech data, and secondly we collected a small development corpus of spoken interactions (labelled `adapt11` here), containing approximately 2,000 words. In development experiments, a portion of this was held out and used as testing data, using five-fold cross-validation.

We restricted the recogniser's vocabulary to the complete set of words from `student09`, plus filled pauses and common contractions such as "it's", "you've" etc. Models were created by linearly interpolating models trained on the three corpora. We also investigated an adaptation technique, which we describe below.

3.1. Adaption of unigram marginals

Given a small set of adaptation data, the unigram probabilities are more reliably estimated than those for higher order n-grams. [10] proposed an adaption scheme to preserve the unigram marginal probabilities estimated from the adaptation data, A, whilst attempting to keep the conditional probabilities as close as possible to the estimates from the background model, B. Adaptation data probabilities and background model probabilities are denoted $p_A(\ldots)$, $p_B(\ldots)$ respectively. Writing $p(w|h)$ for the probability of a word w, given its history h, and $p(w, h)$ for the joint probability of w and h, the scheme attempts to satisfy constraints on the unigram marginals $p_A(w_i)$

$$\sum_h p(w_i, h) = p_A(w_i) \tag{1}$$

for each word w_i in the vocabulary, whilst minimising the KL-divergence with probabilities from the background model, given by:

$$D_{\mathrm{KL}}(p||p_B) = \sum_{h,w} p(w, h) \log \frac{p(w, h)}{p_B(w, h)} \tag{2}$$

$$= \sum_{h,w} p(w|h)p(h) \log \frac{p(w|h)}{p_B(w|h)} \tag{3}$$

The problem is equivalent to finding the maximum entropy distribution satisfying the constraints (1) and may be solved using Generalized Iterative Scaling [11]. Following [10], the adapted

[1] http://htk.eng.cam.ac.uk

conditional probabilities are computed as

$$p(w_i|h_i) = p_B(w_i|h_i)\left(\frac{p_A(w_i)}{p_B(w_i)}\right)^{\beta} \tag{4}$$

where β is a factor controlling the weight of the adaptation data, empirically found in the previous studies to be 0.5.

3.2. Language modelling development results

Results of interpolating trigram language models trained on the three data sets are shown in Table 1. All models were smoothed using Modified Kneser-Ney smoothing. The perplexity is substantially reduced with all model combinations, with the in-domain `student09` data giving the most advantage to models trained solely with the adaptation data. Including speech from `fisher` provides a small additional improvement.

Model	Perplexity
adapt	41.49
fisher	242.36
student09	82.46
student09 + fisher	37.71
adapt + fisher	37.47
adapt + student09	28.51
adapt + student09 + fisher	**27.42**

Table 1: *Perplexity results for interpolated models, evaluated on held-out data from* `adapt11`.

The results of adapting to the unigram marginals of `adapt11` are shown in Table 2. The parameter β was optimised separately for each case. There are consistent improvements over the unadapted interpolated models in each case. We therefore selected the model in the final row of the table for use in the system.

Model	Perplexity
student09 + fisher	31.67
adapt + fisher	34.22
adapt + student09	26.90
adapt + student09 + fisher	**25.45**

Table 2: *Perplexity results for interpolated models with adaptation of unigram marginals.*

4. Acoustic modelling

Due to the limited quantities of development audio data available, we did not attempt to train acoustic models on in-domain data, but instead used models available to us from the AMIDA corpus ([12]), which were trained on approximately 130 hours of speech from multiparty meetings. They are a reasonable match for our domain in terms of the recording conditions, speaking style and speaker demographic. The models were standard HMM-GMMs, trained on PLP features using MPE training. A global HLDA transform was used. Online CMN was performed using ATK's standard method.

4.1. Online speaker adaptation

We considered it important to perform online speaker adaptation in order to improve the ASR performance as quickly as possible. We used CMLLR adaptation [13] for this purpose, which may be formulated as a feature-space transform:

$$\hat{o}_t = Ao_t + b \tag{5}$$

We applied the forward-backward algorithm over the lattices for each utterance created by HDecode, to accumulate CMLLR statistics in an online manner. After returning the one-best hypothesis to the dialogue manager, the system performs background estimation of a new set of transforms using the updated statistics. Sufficient statistics for CMLLR estimation for the GMM are given by:

$$G_i^{(r)} = \sum_{m\in r} \frac{1}{\sigma_i^{(m)2}} \sum_t \gamma_t^{(m)} \begin{bmatrix} 1 & o_t^T \\ o_t & o_t o_t^T \end{bmatrix} \tag{6}$$

$$k_i^{(r)} = \sum_{m\in r} \frac{\mu_i^{(m)}}{\sigma_i^{(m)2}} \sum_t \gamma_t^{(m)} \begin{bmatrix} 1 & o_t^T \end{bmatrix}^T \tag{7}$$

where o_t denotes the observation at time t, $\mu^{(m)}$, $\sigma^{(m)2}$ are the mean and variance of Gaussian m respectively, and $\gamma_t^{(m)}$ denotes the posterior probability of o_t being generated by Gaussian m.

Since it is important to perform adaptation rapidly, we would ideally begin estimating transform sets after as few frames as possible. However, when few frames have been seen the transforms are unlikely to be estimated robustly, and in the worst scenario, this would quickly create a cycle of poor ASR performance, and further poor transform estimation. To avoid this, we investigated the use of a count smoothing technique proposed by [14] to obtain more robust statistics for online operation.

Given prior statistics $G_{\texttt{pri}}^{(r)}$, $k_{\texttt{pri}}^{(r)}$ and a prior weight τ, the final statistics used are

$$G_{\texttt{sm}}^{(r)} = G^{(r)} + \frac{\tau}{\sum_{m\in r}\gamma^{(m)}} G_{\texttt{pri}}^{(r)} \tag{8}$$

$$k_{\texttt{sm}}^{(r)} = k^{(r)} + \frac{\tau}{\sum_{m\in r}\gamma^{(m)}} k_{\texttt{pri}}^{(r)} \tag{9}$$

The prior statistics are obtained by estimating a simpler prior speaker transform and then computing expected statistics in the speaker-specific feature space by applying the prior transform. We investigated the use of two priors transforms: the identity transform, and a transform with the matrix A constrained to be diagonal.

4.2. Online adaptation development results

	WER (%)
no adaption	48.8
full covariance	53.7
block diagonal	45.8
diagonal	47.6

Table 3: *Baseline WER results on the* `adapt11` *set with online adaptation limited to 10 utterances*

We carried out experiments in the use of count smoothing to improve the robustness of online speaker adaptation. Firstly, we artificially limited the amount of data available for transform estimation by resetting the accumulated statistics to zero after every ten utterances. Table 3 show the performance of standard full covariance, block diagonal and diagonal transforms. Two transforms – one for speech and one for silence – were estimated. In all cases we imposed a minimum count threshold on the statistics before estimating a transform. Figure 5 shows the change in performance when full covariance adaptation transforms are smoothed using diagonal and identify prior transforms, for varying prior weight τ. It can be seen that the smoothing gives a clear improvement to the full covariance transforms, reducing the WER to 43.0%, significantly below the results from using a diagonal or block diagonal transform. We found that the identity prior outperformed the diagonal prior for all values of τ.

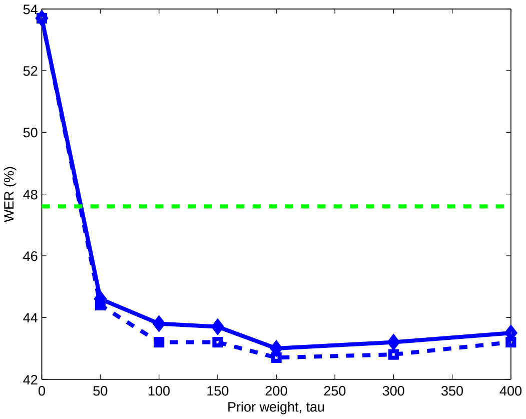


Figure 5: *WER (%) on the* `adapt11` *set using online adaptation limited to 10 utterances. Results are shown with adaptation with diagonal transforms (green dashed); and full covariance transformed smoothed with a diagonal prior (solid blue) and identity prior (dashed blue), with varying prior weight.*

Following these experiments, we applied smoothing with the identity prior to incremental online adaptation on the `adapt11` full data set, where each user's session with the system has around 75 utterances. We first used two transforms, as above, and also investigated the use of regression class trees. Results are shown in Table 4. We found that the smoothing method improves the effectiveness of online adaptation, even when relatively large amounts of data were available for each speaker.

	base	r32
no adaption	48.8	
block diagonal	44.6	41.7
full covariance	41.9	39.2
smoothed	41.6	37.7

Table 4: *WER results (%) for recognition of the full* `dev11` *set with online speaker adaptation on full* `dev11` *set with a transforms for speech and silence (base); and with a regression class tree (r32)*

5. Future work

In this paper we have described the design choices made in implementing ASR for an open-input intelligent tutoring system, but a number of other problems must be solved to create an effective spoken language system. Clearly a major challenge is ensuring robust spoken language understanding when the WER is relatively high, given that the student utterances often have a complex semantic representation. The TRIPS parser is designed to provide robust parses over lattices; however, since the higher-level modules are deterministic in nature, we are not yet able to use the deep domain knowledge available to them to re-score ASR lattices. Furthermore, the parser is tuned to maximise the chance of finding a complete spanning parse, rather than to discriminate between alternative hypotheses. We plan to address this in future work.

Additionally, the system does not yet use statistical dialogue management. We propose to employ reinforcement learning in a future version of the system. Major unsolved issues to consider will be determining a suitable low-dimensional state-space for the dialogue, and selecting which measures of system or student performance should be optimised.

6. References

[1] M. T. H. Chi, N. de Leeuw, M.-H. Chiu, and C. LaVancher, "Eliciting self-explanations improves understanding." *Cognitive Science*, vol. 18, no. 3, pp. 439–477, 1994.

[2] D. Litman, J. Moore, M. Dzikovska, and E. Farrow, "Using natural language processing to analyze tutorial dialogue corpora across domains and modalities," in *Proc. of 14th International Conference on Artificial Intelligence in Education*, 2009.

[3] D. J. Litman and S. Silliman, "ITSPOKE: an intelligent tutoring spoken dialogue system," in *Demonstration Papers at HLT-NAACL 2004*, 2004, pp. 5–8.

[4] H. Pon-Barry, B. Clark, K. Schultz, E. O. Bratt, and S. Peters, "Advantages of spoken language interaction in dialogue-based intelligent tutoring systems," in *Proceedings of ITS-2004*, ser. Lecture Notes in Computer Science, J. C. Lester, R. M. Vicari, and F. Paraguaçu, Eds., vol. 3220. Springer, 2004, pp. 390–400.

[5] N. Person, A. C. Graesser, L. Bautista, E. C. Mathews, and TRG, "Evaluating student learning gains in two versions of AutoTutor," in *Proceedings of AIED-2001*, 2001.

[6] D. Litman, C. P. Rosé, K. Forbes-Riley, K. VanLehn, D. Bhembe, and S. Silliman, "Spoken versus typed human and computer dialogue tutoring," *International Journal of Artificial Intelligence in Education*, vol. 16, pp. 145–170, 2006.

[7] R. G. Hausmann and M. T. Chi, "Can a computer interface support self-explaining?" *Cognitive Technology*, vol. 7, no. 1, pp. 4–14, 2002.

[8] M. Dzikovska, D. Bental, J. D. Moore, N. B. Steinhauser, G. E. Campbell, E. Farrow, and C. B. Callaway, "Intelligent tutoring with natural language support in the Beetle II system," in *Proceedings of ECTEL-2010*. Springer, October 2010, pp. 620–625.

[9] J. Allen, M. Dzikovska, M. Manshadi, and M. Swift, "Deep linguistic processing for spoken dialogue systems," in *Proceedings of the ACL-07 Workshop on Deep Linguistic Processing*, 2007.

[10] R. Kneser, J. Peters, and D. Klakow, "Language model adaptation using dynamic marginals," in *Proc. Eurospeech*, 1997.

[11] J. Darroch and D. Ratcliff, "Generalized iterative scaling for log-linear models," *Annals of Mathematical Statistics*, vol. 43, no. 5, pp. 1470–1480, 1972.

[12] T. Hain, L. Burget, J. Dines, P. Garner, A. el Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, "The AMIDA 2009 meeting transcription system," in *Proc. Interspeech*, 2010, pp. 358–361.

[13] M. Gales, "Maximum likelihood linear transforms for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 75-98, 1998.

[14] C. Breslin, K. Chin, M. Gales, K. Knill, and H. Xu, "Prior information for rapid speaker adaptation," in *Proc. Interspeech*, 2010.