Spoken Language Understanding using the Hidden Vector State Model

Yulan He [a] and Steve Young [b]

[a] *Address for correspondence:*

School of Computer Engineering,

Nanyang Technological University,

Nanyang Avenue, Singapore 639798

Email: asylhe@ntu.edu.sg

Tel: +65 6790 6723 *Fax: +65 6792 6559*

[b] *Cambridge University Engineering Department,*

Trumpington Street, Cambridge CB2 1PZ, England

Email: sjy@eng.cam.ac.uk

Tel: +44 1223 332752 *Fax: +44 1223 332662*

Abstract

The Hidden Vector State (HVS) model is an extension of the basic discrete Markov model in which context is encoded as a stack-oriented state vector. State transitions are factored into a stack shift operation similar to those of a push-down automaton followed by the push of a new preterminal category label. When used as a semantic parser, the model can capture hierarchical structure without the use of treebank data for training and it can be trained automatically using Expectation-Maximization (EM) from only-lightly annotated training data. When deployed in a system, the model can be continually refined as more data becomes available.

In this paper, the practical application of the model in a spoken language understanding system (SLU) is described. Through a sequence of experiments, the issues of robustness to noise and portability to similar and extended domains are investigated. The end-to-end performance obtained from experiments in the ATIS domain show that the system is comparable to existing SLU systems which rely on either hand-crafted semantic grammar rules or statistical models trained on fully-annotated training corpora. Experiments using data which has been artificially corrupted with varying levels of additive noise show that the HVS-based parser is relatively robust, and experiments using data sets from other domains indicate that the overall framework allows adaptation to related domains, and scaling to cover enlarged domains.

In summary, it is argued that constrained statistical parsers such as the HVS model allow robust spoken dialogue systems to be built at relatively low cost, and which can be automatically adapted as new data is acquired both to improve performance and extend coverage.

Keywords: spoken language understanding, spoken dialogue systems, statistical semantic parsing, hidden vector state model

1 Introduction

Robust spoken language understanding (SLU) is a key requirement of spoken dialogue systems. The role of SLU is to robustly interpret the meanings of users' utterances in the face of disturbing effects such as user disfluency and recognition errors. SLU implementation normally comprises three main components: a speech recognizer, a semantic parser to extract the information carrying concepts from the recognized utterance and a dialogue act decoder to determine the overall goal expressed by the utterance. This paper is concerned primarily with the semantic parser.

In most deployed spoken language understanding systems, the semantic parser is based on hand-crafted application-dependent rules. These typically use context-free semantic rules to extract keywords or phrases to fill slots in semantic frames (template matching), examples are MIT's TINA (Seneff, 1992), CMU's PHOENIX (Ward and Issar, 1996), and SRI's Gemini (Dowding et al., 1994). Whilst these hand-crafted approaches can yield good performance, they are expensive to build and they are specific to the application that they were designed for. Furthermore, they can lack robustness when error rates rise or unexpected syntactic forms are used.

In contrast, fully statistical approaches to semantic parsing offer the potential of reduced deployment cost, increased robustness, portability and on-line adaptation to improve and extend domain coverage. Realizing this potential, however, is not straightforward. The primary difficulty is that to construct a model with the expressive power of context-free parsing rules such as in the hierarchical Hidden Understanding Model (Miller et al., 1995; Schwartz et al., 1997) or the hierarchical HMM (Fine et al., 1998) requires in practice fully-annotated tree-bank style train-

ing data and this is expensive to generate. Relaxing the requirement for all training data to be fully observed implies that the exact parse state for every word is no longer known. In effect the parse state becomes a hidden variable which must be estimated along with the model parameters using Expectation-Maximization (EM) style iterative training, for example, in the form of the Inside-Outside algorithm (Lari and Young, 1990; Schabes, 1991). Unfortunately, however, context-free models provide too many degrees of freedom and in practice, models trained in this way do not seem to converge on useful solutions (Lari and Young, 1991). One can of course use finite-state parsing models such as that used in AT&T's Markov model-based CHRONUS (Levin and Pieraccini, 1995) but this results in essentially a HMM-based semantic tagger which although capable of being robustly trained, is not capable of representing hierarchical structure in the data.

Recently the authors have proposed a hidden vector state (HVS) model which is essentially a stochastic push-down automaton. The HVS model extends the basic discrete Markov model by expanding each state to encode the stack of a push-down automaton. This allows the model to efficiently encode hierarchical context, but because stack operations are highly constrained it avoids the tractability issues associated with full context-free stochastic models such as the hierarchical HMM. This model is capable of representing right-branching context-free grammars and can be robustly trained using EM (He and Young, 2003b). The model has been tested in an ATIS-based spoken language system and shown to be comparable in performance to hand-crafted systems but without any hand-crafted rules or application dependent tuning (He and Young, 2003a).

This paper explores the performance of the HVS model when used in a spoken language understanding system (He and Young, 2004). Following a brief description of the system a series of experiments are presented which explore the robustness of

the model to noise and the portability of the model to similar and extended domains.

The structure of the paper is as follows. In section 2 the overall system architecture is described. Then in section 3 the "end-to-end" performance is measured using the ATIS evaluation framework. These results demonstrate that the system is comparable to the original DARPA ATIS SLU systems without using any hand-crafted rules or system tuning. In section 4, noise robustness is investigated by testing the system on data from the ATIS domain which has been artificially corrupted with varying levels of additive noise. Then in section 5, portability issues are explored by adapting an HVS model originally trained on the ATIS corpus to the DARPA Communicator task which covers broadly similar concepts, but comprises rather different speaking styles. Extension of the ATIS-trained HVS model to an enlarged domain which includes Tourist Information queries is also presented where in this case many unseen semantic concepts have been introduced. Finally, Section 6 concludes the paper.

2 The HVS-based Statistical SLU System

The SLU system architecture is shown in Fig. 1. It consists of a speech recognizer, a semantic parser, and a dialog act decoder. The speech recognizer processes each input acoustic signal A to produce the N-best word string hypotheses W_n. The semantic parser then determines a parse for each W_n, extracts a set of semantic concepts C and computes the associated probability $P(C|W_n)$. The most likely interpretation is then given by the joint optimization over C and W_n

$$\hat{C} = \underset{C}{\text{argmax}} \left\{ \max_{W_n;\, n \in 1:N} P(C|W_n) P(W_n|A) \right\} \tag{1}$$

This joint optimization avoids the error incurred by sequentially decoding $\hat{W}$ and then $\hat{C}$ (Young, 2002).

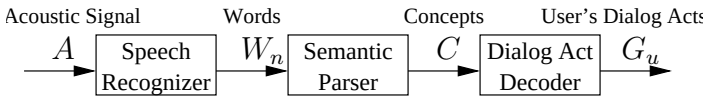


Fig. 1. Typical structure of a spoken language understanding system.

Given the set of semantic concepts $\hat{C}$ the dialogue act decoder infers the user's dialog acts or goals by solving [1]

$$\hat{G_u} = \underset{G_u}{\text{argmax}}\ P(G_u|\hat{C}) \tag{2}$$

In the system described in this paper, each of these stages is modeled separately. We use a standard HTK-based Hidden Markov Model (HMM) recognizer for speech recognition (Young et al., 2004). The recognizer comprises 14 mixture Gaussian HMM state-clustered cross-word triphones augmented by using heteroscedastic linear discriminant analysis (HLDA) (Kumar, 1997). During decoding, incremental speaker adaptation based on maximum likelihood linear regression (MLLR) (Gales and Woodland, 1996) is performed with updating being performed every five input utterances.

The core of the system is the semantic parser which computes a hierarchical parse tree for each word string W, and then extracts semantic concepts C from this tree. Each semantic concept consists of a name-value pair where the name is a dotted list of primitive semantic concept labels. For example, the top part of figure 2 shows a typical semantic parse tree and the semantic concepts extracted from this parse

[1] It is also possible to retain the N-best parse results from the semantic parser and leave the selection of the best hypothesis until the dialog act decoding stage. However, in practice, no gain was found for this and hence we do not pursue it further here.

would be

```
RETURN.TOLOC.CITY=Dallas
                                    (3)
RETURN.ON.DATE=Thursday
```

The semantic parser is based on the Hidden Vector State (HVS) model (He and Young, 2003b, 2005) which is a discrete Hidden Markov Model (HMM) in which each HMM state represents the state of a push-down automaton with a finite stack size. This is illustrated in figure 2 which shows the sequence of HVS stack states corresponding to the given parse tree. State transitions are factored into separate stack pop and push operations constrained to give a tractable search space. The result is a model which is complex enough to capture hierarchical structure but which can be trained automatically from only lightly annotated data.

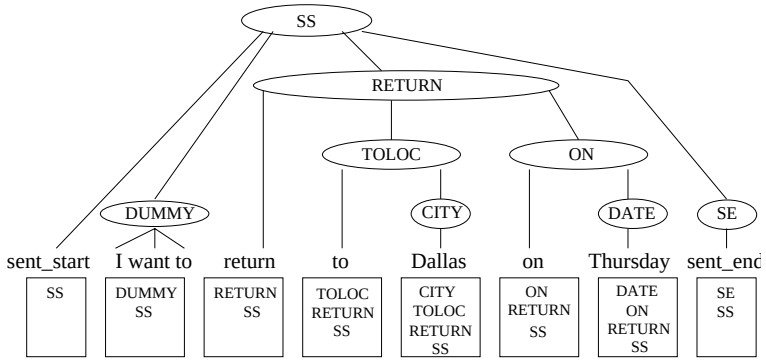


Fig. 2. Example of a parse tree and its vector state equivalent.

In the HVS-based semantic parser, conventional rules are replaced by three probability tables. Let each state at time t be denoted by a vector of D_t semantic concept labels (tags) $\mathbf{c}_t = [c_t[1], c_t[2], ..c_t[D_t]]$ where $c_t[1]$ is the preterminal concept label and $c_t[D_t]$ is the root concept label (SS in Figure 2). Given a word sequence W, concept vector sequence $\mathbf{C}$ and a sequence of stack pop operations N, the joint probability

of $P(W, \mathbf{C}, N)$ can be decomposed as

$$P(W, \mathbf{C}, N) = \prod_{t=1}^{T} P(n_t|\mathbf{c}_{t-1}) P(c_t[1]|c_t[2 \cdots D_t]) P(w_t|\mathbf{c}_t) \quad (4)$$

where n_t is the vector stack shift operation and takes values in the range $0, \cdots, D_{t-1}$, and $c_t[1] = c_{w_t}$ is the new pre-terminal semantic label assigned to word w_t at word position t.

Thus, the HVS model consists of three types of probabilistic move, each move being determined by a discrete probability table:

(1) popping semantic labels off the stack - $P(n|\mathbf{c})$;
(2) pushing a pre-terminal semantic label onto the stack - $P(c[1]|c[2 \cdots D])$;
(3) generating the next word - $P(w|\mathbf{c})$.

Each of these tables are estimated in training using EM and then used to compute parse trees at run-time using Viterbi decoding. In training, each word string W is marked with the set of semantic concepts $\mathbf{C}$ that it contains. For example, if the sentence shown in figure 2 was in the training set, then it would be marked with the two semantic concepts given in 3. For each word w_k of each training utterance W, EM training uses the forward-backward algorithm to compute the probability of the model being in stack state $\mathbf{c}$ when w_k is processed. Without any constraints, the set of possible stack states would be intractably large. However, in the HVS model this problem can be avoided by pruning out all states which are inconsistent with the semantic concepts associated with W. The details of how this is done are given in (He and Young, 2005).

The dialog act decoder uses the Bayesian Network approach proposed in (Meng et al., 2000) extended to use Tree-Augmented Naive Bayes (TAN) networks (Friedman et al., 1997) in order to model between-concept dependencies. One TAN net-

work is used for each dialogue act or goal G_u, the set of semantic concept labels $\{c_i\}$ which serve as input to its corresponding network were selected based on the mutual information (MI) between the goal and each concept label. Naive Bayes networks assume all the concept labels are conditionally independent given the value of the goal. TAN networks relax this independence assumption by adding dependencies between concept labels based on the conditional mutual information (CMI) between concepts given the goal. The goal prior probability $P(G_u)$ and the conditional probability of each semantic concept label c_i given the goal G_u, $P(c_i|G_u)$ are learned from the training data. Dialogue act detection is done by selecting the goal with the highest posterior probability of G_u given the particular instance of concepts $c_1 \cdots c_n$, $P(G_u|c_1 \cdots c_n)$.

3 End-to-End Performance

The end-to-end performance of the HVS-based SLU system was measured using the Air Travel Information Services (ATIS) corpus (Dahl et al., 1994). ATIS was developed in the DARPA sponsored spoken language understanding programme conducted from 1990 to 1995 and it provides a convenient and well-documented standard for measuring the end-to-end performance of an SLU system.

3.1 Experimental Setup

The three components of the SLU system, the speech recognizer, the semantic parser, and the dialogue act decoder were trained separately. For the HTK-based speech recognizer, 22316 spontaneous utterances recorded using a Sennheiser microphone from ATIS-2 and ATIS-3 were used for acoustic model training. This includes the

ATIS-2 FEB92 and NOV92 test sets in addition to the ATIS-2 and ATIS-3 training sets. The statistical language model was trained on 23096 ATIS spontaneous utterances with a vocabulary size of 1644 words. It consists of a word trigram interpolated with a class-based trigram. The latter has 60 classes derived automatically using the Kneser-Ney clustering procedure (Kneser and Ney, 1993). The perplexity tested on the joint ATIS-3 NOV93 and DEC94 test sets is 15.5. If the word trigram is used alone, the perplexity increases to 16.5.

As explained in section 2, the N-best word hypotheses W_n generated by the speech recognizer are processed by the semantic parser in order to perform a joint optimization over words W and concepts C. In practice, equation 1 is modified to allow the relative likelihoods computed by the recognizer's statistical language model and the parser's semantic model to be balanced:

$$\begin{aligned}\hat{C} &= \underset{C}{\text{argmax}} \left\{ \max_{W_n;\, n\in 1:N} P(C|W_n)P(W_n)P(A|W_n) \right\} \\ &\approx \underset{C}{\text{argmax}} \left\{ \max_{W_n;\, n\in 1:N} P(C|W_n)^{\alpha}P(W_n)^{\gamma}P(A|W_n) \right\} \end{aligned} \tag{5}$$

where $P(A|W_n)$ is the acoustic probability computed by the recognizer, $P(W_n)$ is the language model likelihood, and $P(C|W_n)$ is the semantic parse likelihood. The weighting factors α and γ were set empirically to optimize performance. In the case of the DEC94 test, a development set was used and values of $\alpha = 10$ and $\gamma = 17$ were found to optimize performance. In the case of the NOV93, no development set was available and values of $\alpha = 10$ and $\gamma = 15$ were determined directly using the test set. Although this biased the NOV93 results, later analysis showed that the overall results were not in fact sensitive to the exact setting of these values.

The HVS semantic parser was trained on 4978 utterances selected from the Class A training data in ATIS-2 and ATIS-3. Each utterance was marked with the concepts

it contains as described in section 2. These were derived automatically from the SQL queries accompanying the ATIS training utterances. The semantic class labels were also derived automatically from the domain-specific lexical class information in the ATIS database to give 30 semantic classes in all. Further semantic classes were then derived from the ATIS SQL query set such as FROMLOC, TOLOC, ARRIVE_DATE, DEPART_DATE etc. In total, 85 semantic concepts were defined.

For the dialog act decoder, 16 dialog acts or goals were defined in the ATIS domain by enumerating the key attribute to be fetched from each SQL query with each goal corresponding to one TAN. Examples of the dialogue acts defined are abbreviation, airfare, flight etc. The top 15 semantic concepts ranked by a mutual information metric were used as input to each TAN.

In order to test performance within the ATIS framework, the output of the dialog act decoder must be combined with the extracted semantic concepts to form an SQL query. The SQL query generator module was tested on the reference parse results of ATIS-3 NOV93 and DEC94 test sets. 5 out of 448 utterances from NOV93 test set and 3 out of 445 utterances from DEC94 test set did not return the correct answers, which gives the utterance understanding error rate 1.1% and 0.7% respectively. The analysis of the results shows that one context-dependent utterance has been misclassified as category A (context-independent) in each of these two test sets and the rest are too complicated for the SQL query generator to handle properly.

3.2 Experimental Results

The end-to-end performance evaluation results on both natural language understanding (NL) and spoken language understanding (SLS) evaluations are shown in

Table I. F-measure evaluates the extraction of concept/value pairs in terms of recall and precision (Goel and Byrne, 1999), while answer error rate measures the minimum/maximum answers from the ATIS database using the NIST scoring package. The latter is the standard scoring metric used by DARPA ATIS SLU systems. For the NL test, the reference transcription is used as input to the semantic parser instead of the recognized output. The SLS(1) results were obtained by using only the best hypothesis from the speech recognizer, while the SLS(10) results were obtained by jointly optimizing over the top 10 hypotheses output by the recognizer.

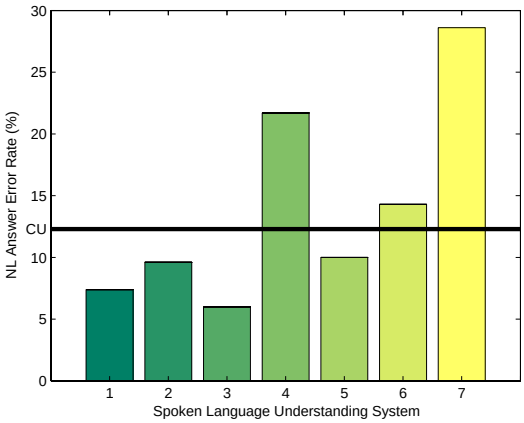
It can be observed from Table I that the joint optimization reduces the WER by 7.3% and 10.0% relatively for the NOV93 and DEC94 test set respectively. The relative reduction in answer error rate is 12.0% and 9.4%.

Table I

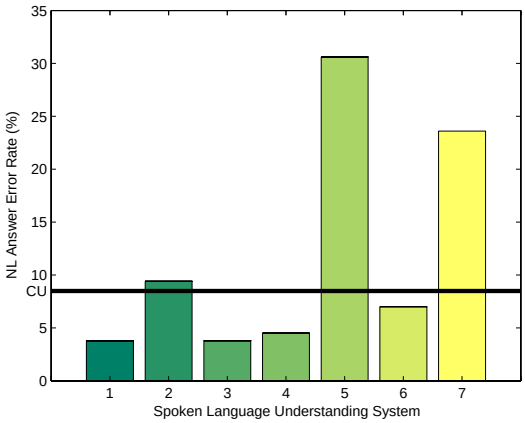
NOV93 and DEC94 NL and SLS test results.

	NOV93			*DEC94*		
	WER	*F-measure*	*Answer Error*	*WER*	*F-measure*	*Answer Error*
NL	-	90.3%	12.3%	-	91.9%	8.5%
SLS(1)	4.1	89.0%	18.3%	3.0	90.5%	13.9%
SLS(10)	3.8	89.3%	16.1%	2.7	90.6%	12.6%

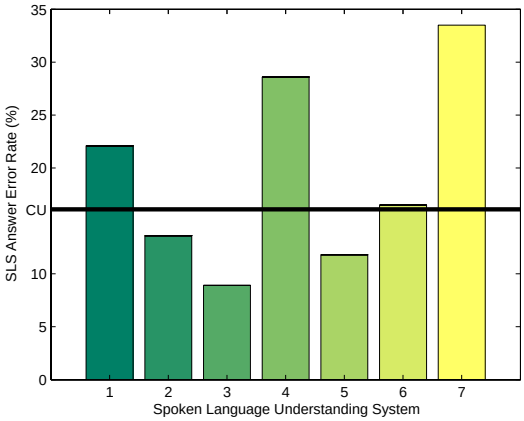
Figure 3 compares the performance of the system built here (denoted as CU in y-axis) with the systems developed by the DARPA ATIS programme participants. The bar chart shows the performance of other systems and the solid line across the bar chart illustrates our system's performance. The upper portion of Figure 3 gives the



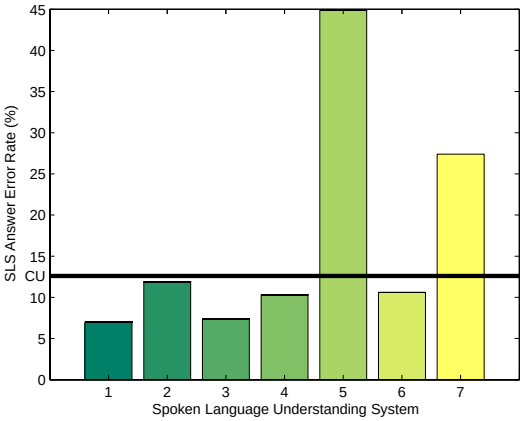
(a) NOV93 NL Answer Error Rate



(c) DEC94 NL Answer Error Rate



(b) NOV93 SLS Answer Error Rate



(d) DEC94 SLS Answer Error Rate

Fig. 3. SLU Systems Performance Comparison.

NL answer error rates of various systems on the NOV93 and DEC94 test sets, while the lower portion of the figure gives the SLS answer error rates on the same test sets. It can be observed that the statistical SLU system described here is comparable to the original DARPA ATIS SLU systems despite having no hand-crafted rules and no tuning. [2]

[2] Note also that a number of the DARPA ATIS systems used other corpora to augment their training sets.

4 Noise Robustness

Spoken language understanding components should be robust to speech recognition errors. Ideally, they should be capable of generating the correct meaning of an utterance even if it is recognized wrongly by a speech recognizer. At minimum, the performance of the understanding components should degrade gracefully as recognition accuracy degrades.

To test the robustness of the HVS-based system, varying levels of car noise from the NOISEX-92 (Varga et al., 1992) database was added to the ATIS-3 NOV93 and DEC94 test sets. The resulting noisy speech utterances had signal-to-noise ratios (SNRs) varying from "clean" down to 10dB.

Performance was measured at both the component and the system level. For the former, the recognizer was evaluated by word error rate, the parser by concept slot retrieval rate using an F-measure metric, and the dialog act decoder by detection rate. The overall system performance was measured using the standard NIST "query answer" rate.

4.1 Experimental Results

Figure 4 gives the system performance on the corrupted test data with additive noise ranging from 25dB to 10dB SNR. The label "clean" in the X-axis denotes the original clean speech data without additive noise. Note that the recognition results on the corrupted test data were obtained directly using the original clean speech HMM models without retraining for the noisy conditions or other forms of noise compensation. The upper portion of Figure 4 shows the end-to-end performance

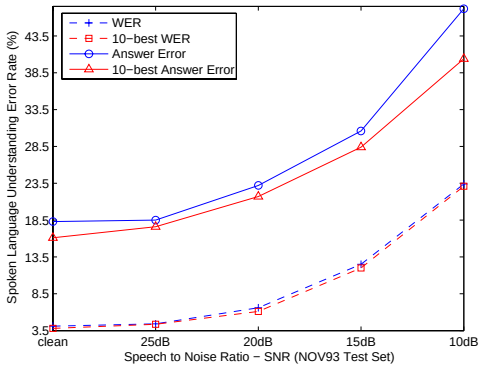
in terms of query answer error rate for the NOV93 and DEC94 test sets. For easy reference, WER is also shown. The individual component performance, F-measure for the HVS semantic parser and dialogue act (DA) detection accuracy for the DA decoder, are illustrated in the lower portion of Figure 4. For each test set, the performance is given for the single-best recognizer output (i.e. $N = 1$ in equation 5) and for the jointly optimized case ($N = 10$ in equation 5). The latter are designated as "10-best" in the figures.

It can be observed that the system gives fairly stable performance at high SNRs and then the recognition accuracy degrades rapidly in the presence of increasing noise. At 20dB SNR, the WER for the NOV93 test set increases by 1.6 times relative to clean whilst the query answer error rate increases by only 1.3 times. On decreasing the SNR to 15dB, the system performance degrades significantly. The WER increases by 3.1 times relative to clean but the query answer error rate increases by only 1.7 times. Similar figures were obtained for the DEC94 test set.

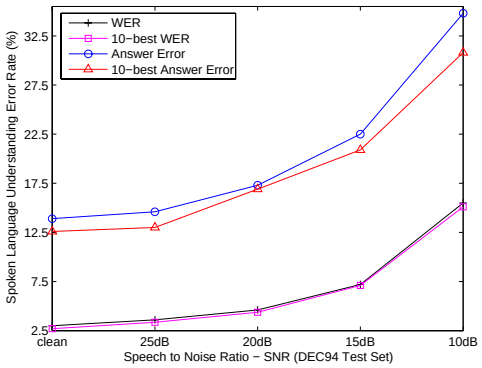
The above suggests that the end-to-end performance measured in terms of answer error rate degrades more slowly compared to the recognizer WER as the noise level increases. This demonstrates that the statistically-based understanding components of the SLU system, the semantic parser and the dialogue act decoder, are relatively robust to degrading recognition performance.

Regarding the individual component performance, the dialogue act detection accuracy appears to be less sensitive to decreasing SNR. This is probably a consequence of the fact that the Bayesian networks are set up to respond to only the presence or absence of semantic concepts or slots, regardless of the actual values assigned to them. In another words, the performance of the dialogue act decoder is not affected by the mis-recognition of individual words, but only by a failure to detect the pres-

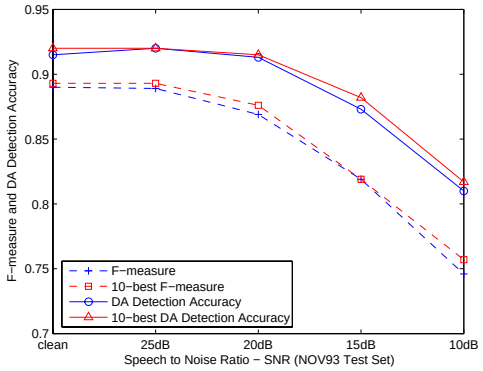
ence of a semantic concept. It can also be observed from Figure 4 that the overall answer error rate increases steeply below 15dB from which it can be inferred that, as a very rough guide, the concept slot F-measure needs to be better than around 85% in order to achieve acceptable end-to-end performance.



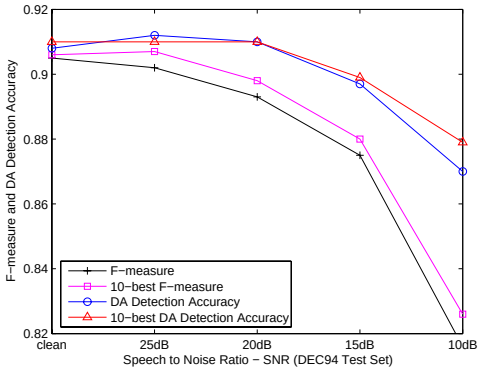
(a) NOV93 End-to-End Performance



(c) DEC94 End-to-End Performance



(b) NOV93 Component Performance



(d) DEC94 Component Performance

Fig. 4. SLU system performance vs SNR.

5 Portability and Extensibility

To test the portability and extensibility of the statistical parser, two sets of experiments have been conducted based on two different scenarios. In scenario one,

the ATIS-trained HVS model was tested in a domain which covers broadly similar concepts, but comprises rather different speaking styles. To this end, the flight information subset of the DARPA Communicator Travel task was used as the target domain (CUData, 2004). In scenario two, the ability to extend domain coverage to include queries relating to tourist information was studied. Data for this extended domain were obtained from the SACTI corpus (Williams and Young, 2004; Stuttle et al., 2004) and in this case, a large number of new semantic concepts are needed.

For reference, some statistics of the ATIS, DARPA Communicator, and SACTI corpora are given in Table II.

Table II

Corpus statistics for the HVS semantic parser.

	ATIS	*DARPA Communicator*	*SACTI*
Training Set (No. of Utt.)	4978	10682	1621
Test Set (No. of Utt.)	993	1017	157
Vocabulary Size	611	505	786
Semantic Concept Labels	85	99	80

Note that the semantic concept labels needed for ATIS form a subset of those needed for Communicator. For SACTI, only 14 of the 80 semantic concept labels appear in the ATIS set.

5.1 Adaptation to Changing Domains

Statistical model adaptation techniques are widely used to reduce the mismatch between training and test or to adapt a well-trained model to a novel domain. Commonly used techniques can be classified into two categories, Bayesian adaptation which uses a maximum *a posteriori* (MAP) probability criteria (Gauvain and Lee, 1994) and transformation-based approaches such as maximum likelihood linear regression (MLLR) (Gales and Woodland, 1996), which uses a maximum likelihood (ML) criteria. In recent years, MAP adaptation has been successfully applied to n-gram language models (Bacchiani and Roark, 2003) and lexicalized PCFG models (Roark and Bacchiani, 2003). Luo *et al.* have proposed transformation-based approaches based on the Markov transform (Luo et al., 1999) and the Householder transform (Luo, 2000), to adapt statistical parsers. However, the optimization processes for the latter are complex and it is not clear how general they are.

Since MAP adaptation is straightforward and has been applied successfully to PCFG parsers, it has been selected for investigation in this paper. Since one of the special forms of MAP adaptation is interpolation between the in-domain and out-of-domain models, it is natural to also consider the use of non-linear interpolation and hence this has been studied as well [3].

5.1.1 MAP Adaptation

Bayesian adaptation re-estimates model parameters directly using adaptation data. It can be implemented via maximum *a posteriori* (MAP) estimation. Assuming that

[3] Experiments using linear interpolation have also been conducted but it was found that the results are worse than those obtained using MAP adaptation or log-linear interpolation.

model parameters are denoted by Θ, then given observation samples Y, the MAP estimate is obtained as

$$\Theta_{MAP} = \underset{\Theta}{\text{argmax}} P(\Theta|Y) = \underset{\Theta}{\text{argmax}} P(Y|\Theta)P(\Theta) \tag{6}$$

where $P(Y|\Theta)$ is the likelihood of the adaptation data Y and model parameters Θ are random vectors described by their probabilistic mass function (pmf) $P(\Theta)$, also called the prior distribution.

In the case of HVS model adaptation, the objective is to estimate probabilities of discrete distributions over vector state stack shift operations and output word generation. Assuming that they can be modeled as multinomial distributions, the Dirichlet density can be used as conjugate prior. Then given a parser model $P(W, C)$ for a word sequence W and semantic concept sequence C with J component distributions P_j each of dimension K, and given some adaptation data W_l, the MAP estimate of the kth component of P_j, $\hat{P}_j(k)$, is

$$\hat{P}_j(k) = \frac{\sigma_j}{\sigma_j + \tau}\tilde{P}_j(k) + \frac{\tau}{\sigma_j + \tau}P_j(k) \tag{7}$$

where $\sigma_j = \sum_{k=1}^{K} \sigma_j(k)$ in which $\sigma_j(k)$ is defined as the total count of the events associated with the kth component of P_j summed across the decoding of all adaptation utterances W_l, $P_j(k)$ is the probability of the original unadapted model, and $\tilde{P}_j(k)$ is the empirical distribution of the adaptation data, which is defined as

$$\tilde{P}_j(k) = \frac{\sigma_j(k)}{\sum_{i=1}^{K} \sigma_j(i)} \tag{8}$$

τ is the prior weighting parameter and this was optimized empirically on a held-out set.

As discussed in section 2, the HVS model consists of three types of probabilistic move. The MAP adaptation technique can be applied to the HVS model by adapting

each of these three component distributions individually.

5.1.2 Log-Linear Interpolation

Log-linear interpolation has been applied to language model adaptation and has been shown to be equivalent to a constrained minimum Kullback-Leibler distance optimization problem (Klakow, 1998).

Using the same notation as in section 5.1.1, the log-linear form of adaptation can be written as

$$\hat{P}_j(k) = \frac{1}{Z_\lambda} P_j(k)^{\lambda_1} \tilde{P}_j(k)^{\lambda_2} \tag{9}$$

The parameters λ_1 and λ_2 were determined by optimizing the log-likelihood on the held-out data using the simplex method. The computation of Z_λ is very expensive and can usually be dropped without significant loss in performance (Martin et al., 2000).

5.1.3 Experiments

The HVS-parser model trained on the ATIS data was adapted using utterances relating to flight reservation from the DARPA Communicator data. To compare the adapted ATIS parser with an in-domain Communicator parser, a HVS model was trained from scratch using 10682 Communicator training utterances. For all tests, a set of 1017 Communicator test utterances was used.

Table III lists the recall, precision, and F-measure results obtained when tested on the DARPA Communicator test set. The baseline is the unadapted HVS parser trained on the ATIS corpus only. The in-domain results are obtained using the HVS parser trained solely on the 10682 DARPA training data. The other rows of the

table give the parser performance using MAP and log-linear interpolation based adaptation of the baseline model using 50 randomly selected adaptation utterances.

Table III

Performance comparison of adaptation to DARPA Communicator Data using MAP or log-linear interpolation.

System	*Recall*	*Precision*	*F-measure*
Baseline	79.8%	87.1%	83.3%
In-domain	87.2%	91.9%	89.5%
MAP	86.7%	91.1%	88.9%
Log-Linear	86.3%	92.4%	89.2%

Since a reference database for the DARPA Communicator task was not available, it was not possible to conduct an end-to-end performance evaluation as in section 3. However, the experimental results in section 4.1 suggested that the F-measure needs to exceed 85% to give acceptable end-to-end performance (see Figure 4). Therefore, it can be inferred from Table III that the unadapted ATIS parser model would perform rather badly in the new Communicator application whereas the adapted models would give performance close to that of a fully trained in-domain model.

Figure 5 shows the parser performance versus the number of adaptation utterances used. It can be observed that when there are only a few adaptation utterances, MAP adaptation performs significantly better than log-linear interpolation. However above 25 adaptation utterances, the converse is true. The parser performance saturates when the number of adaptation utterances reaches 50 for both techniques and the best performance overall is given by the parser adapted using log-linear interpolation.

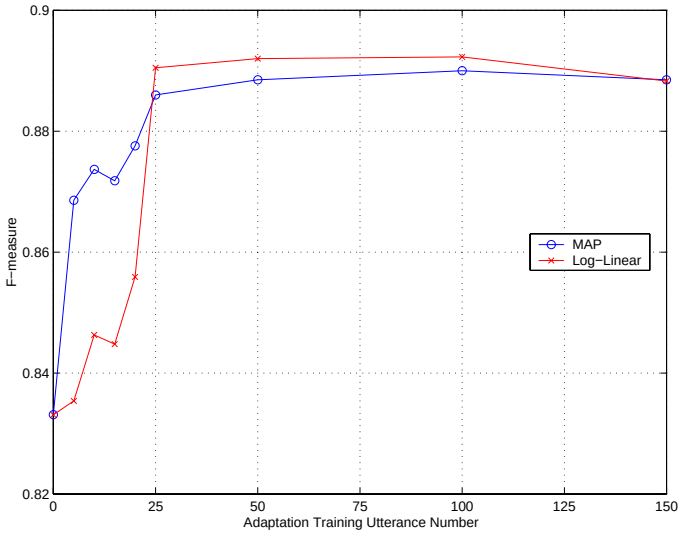


Fig. 5. F-measure vs amount of adaptation training data.

5.2 Extension to Expanded Domains

In this section, preliminary results are reported using the first phase of a new dialogue dataset called the Simulated ASR Channel Tourist Information (SACTI) corpus (Williams and Young, 2004; Stuttle et al., 2004). The SACTI corpus consists of task-oriented dialogues between two people in a simulated ASR channel. This channel uses a phonetic confusion model and a language model to simulate recognition errors in the range 0% WER to 60% WER. The corpus contains a total of 144 dialogues, within which, 132 dialogues (1621 utterances) were chosen as the training set and the remaining 12 dialogues (157 utterances) formed the test set. Since the error rates in some of the dialogues are very high, the user utterances include a mixture of queries, repairs and clarifications. For the work described here, the error-free user utterances prior to scrambling are used. Nevertheless, this remains a very demanding understanding task.

The focus of the experiments was to investigate how effectively the HVS model could be extended to cover an enlarged domain, in this case the union of the ATIS and

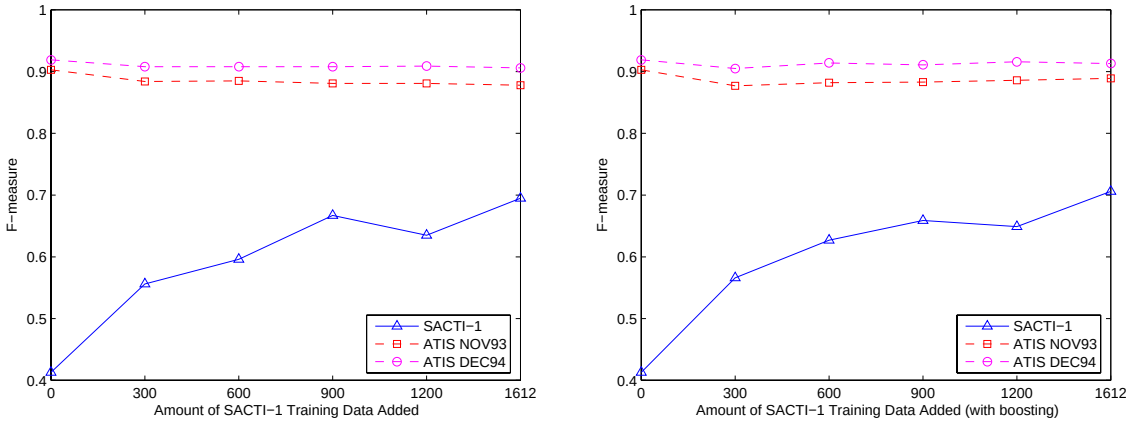
SACTI domains. Two approaches have been tested: training a combined model on the pooled ATIS and SACTI training data, and interpolating individual ATIS and SACTI domain models.

Of the 85 concept labels needed to cover the ATIS domain, only 14 are shared with the 80 concepts needed to cover the SACTI domain. Thus, a combined model simultaneously covering both domains requires 151 distinct concept labels. Examples of the new semantic concept labels in the SACTI corpus include FILM, BAR_NAME, BUS_NUMBER, etc. The shared concept labels include COST, DAY_NAME, DISTANCE, FROMLOC, TOLOC, QUANTITY, TIME, etc.

A HVS model trained only on the SACTI data results in an F-measure of 73.3% on the SACTI test set. This relatively poor figure is a consequence of the small training set and the difficulty of the task.

The results of training a combined model are shown in Figure 6 which shows the F-measure obtained on both the ATIS and SACTI test sets. The horizontal axis shows the effect of adding varying amounts of SACTI data to the 4978 utterance ATIS training set. Figure 6 (a) depicts the results without boosting the SACTI data (i.e. the model was trained on 1 x ATIS data + 1 x SACTI data) while Figure 6 (b) shows the results by boosting the SACTI data (i.e. the model was trained on 1 x ATIS data + 3 x SACTI data). It can be observed from this figure that the original ATIS model gives very poor performance on the SACTI test set, only 41.3% F-measure was obtained. However, by incorporating increasing amounts of SACTI training data, the F-measure increased gradually to 69.5% without boosting or 70.6% with boosting. At this point, the combined model is simultaneously covering both domains but with a reduction of approximately 2% F-measure on the ATIS test sets and 4% on the SACTI test set. By boosting the SACTI data, the reduction on the

SACTI test set is decreased to 3%.



(a) Without Boosting.

(b) With Boosting.

Fig. 6. F-measure vs amount of SACTI training data added.

For interpolating individual domain models, the log-linear scheme used in section 5.1.3 is not appropriate since one domain cannot be regarded as a prior for the other. Instead, simple linear interpolation of the individual domain models was used and the results are illustrated in Figure 7. In this case, a mixture weight of between 0.6 and 0.7 gives a similar approximate 2% performance drop-off on the ATIS test set however the drop-off on the SACTI test set is reduced to less than 2%. This suggests that model interpolation schemes may be the better approach towards expanding parser coverage, especially where there is an imbalance in the amount of available training data.

Finally it is interesting to explore whether concepts from different domains play sufficiently similar roles, that they can be tied into equivalence classes for the purposes of parameter estimation. As a very preliminary test, a number of equivalence sets were constructed manually by pairing concept labels which appeared in similar contexts in SACTI and ATIS. The effect of tying each pair was then determined

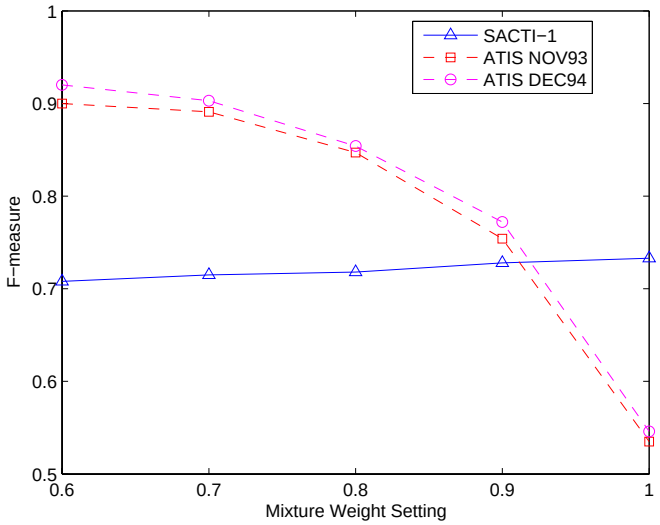


Fig. 7. F-measure vs mixture weight setting (mixture of the ATIS Model and the SACTI model).

individually by retraining the model with the corresponding model parameters tied. The tying of AIRPORT from the ATIS domain with RESTAURANT from the SACTI domain yielded a small increase in F-measure of 1.5% but all other tyings resulted in a small decrease in performance. This remains a topic for future work.

6 Conclusions

The Hidden Vector State (HVS) model is an extension of the basic discrete Markov model in which each state represents the stack of a push-down automaton. State transitions are factored into a stack shift operation followed by the push of a new pre-terminal semantic category label. When used as a semantic parser, the model can be trained directly from only-lightly annotated data whilst at the same time being able to capture hierarchical structure in the data. The use of the HVS model in semantic parsing has been presented previously (He and Young, 2003b, 2005). In this paper, the practical application of a HVS-based parser within a speech understanding

system (SLU) has been described and studied experimentally in terms of system integration issues, robustness to noise, and portability to new and extended domains.

The integrated SLU system presented here is entirely data-driven. The system is trained from data and there are no heuristic rules. Experiments have been conducted on the ATIS corpus and the results show that the system is comparable to existing SLU systems which rely on either hand-crafted semantic grammar rules or statistical models trained on fully-annotated training corpora but it has greatly reduced build cost.

To investigate noise robustness, a set of experiments have been conducted where the acoustic test data was corrupted with varying levels of additive car noise. The end-to-end system performance was then measured along with the individual component performances. It was found that although the addition of noise had a substantial effect on the word error rate, its relative influence on both the semantic parser concept retrieval rate and the dialogue act detection accuracy was somewhat less. Overall, the end-to-end error rate degraded relatively more slowly than word error rate and perhaps most importantly of all, there was no catastrophic failure point at which the system effectively stopped working, a situation not uncommon in current rule-based systems.

The flexibility of the HVS model to adapt to changing domains and extend to cover wider domains has been explored experimentally via two sets of experiments. Firstly, a model trained on ATIS data was adapted to the DARPA Communicator domain. The latter entails a similar set of concepts but with different user speaking styles and different syntactic forms. It was found that applying the ATIS-trained system to Communicator resulted in a 6% absolute drop in F-measure on concept accuracy (i.e. a 62% relative increase in parser error). However, when log-linear adaptation

was applied using only 50 adaptation sentences, the loss in concept accuracy was essentially restored. In the second set of experiments, a HVS model was required to cover the union of two quite different domains. In this case, it was shown that linear interpolation of individual domain models provided a simple but effective solution.

Overall, these results show that constrained statistical parsers such as the HVS model allow robust spoken dialogue systems to be built at relatively low cost, and which can be automatically adapted as new data is acquired both to improve performance and extend coverage.

References

Bacchiani, M., Roark, B., Apr. 2003. Unsupervised language model adaptation. In: Proc. of the IEEE Intl. Conf. on Acoustics, Speech and Signal Processing. Hong Kong.

CUData, 2004. DARPA Communicator Travel Data. University of Colorado at Boulder, http://communicator.colorado.edu/phoenix.

Dahl, D., Bates, M., Brown, M., Hunicke-Smith, K., Pallett, D., Pao, C., Rudnicky, A., Shriberg, L., Mar. 1994. Expanding the scope of the ATIS task: the ATIS-3 corpus. In: ARPA Human Language Technology Workshop. Princeton, NJ.

Dowding, J., Moore, R., Andry, F., Moran, D., June 1994. Interleaving syntax and semantics in an efficient bottom-up parser. In: Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics. Las Cruces, New Maxico, pp. 110–116.

Fine, S., Singer, Y., Tishby, N., 1998. The hierarchical hidden markov model: Analysis and applications. Machine Learning 32, 41–62.

Friedman, N., Geiger, D., Goldszmidt, M., 1997. Bayesian network classifiers. Ma-

chine Learning 29 (2), 131–163.

Gales, M., Woodland, P., Oct. 1996. Mean and variance adaptation within the MLLR framework. Computer Speech and Language 10, 249–264.

Gauvain, J., Lee, C.-H., 1994. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. IEEE Trans. on Speech and Audio Processing 2 (2), 291–298.

Goel, V., Byrne, W., 1999. Task dependent loss functions in speech recognition: Application to named entity extraction. In: ESCA ETRW Workshop on Accessing Information from Spoken Audio. Cambridge, UK, pp. 49–53.

He, Y., Young, S., Dec. 2003a. A data-driven spoken language understanding system. In: IEEE Automatic Speech Recognition and Understanding Workshop. St. Thomas, U.S. Virgin Islands.

He, Y., Young, S., Apr. 2003b. Hidden vector state model for hierarchical semantic parsing. In: Proc. of the IEEE Intl. Conf. on Acoustics, Speech and Signal Processing. Hong Kong.

He, Y., Young, S., May 2004. Robustness issues in a data-driven spoken language understanding system. In: HLT/NAACL Workshop on Spoken Language Understanding for Conversational Systems. Boston, MA.

He, Y., Young, S., 2005. Semantic processing using the hidden vector state model. Computer Speech and Language 19 (1), 85–106.

Klakow, D., Nov. 1998. Log-linear interpolation of language models. In: Proc. of Intl. Conf. on Spoken Language Processing. Sydney, Australia.

Kneser, R., Ney, H., 1993. Improved clustering techniques for class-based statistical language modelling. In: Proceedings of the European Conference on Speech Communication and Technology. pp. 973–976.

Kumar, N., 1997. Investigation of silicon auditory models and generalization of linear discriminant analysis for improved speech recognition. Ph.D. thesis, Johns

Hopkins University, Baltimore MD.

Lari, K., Young, S., 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. Computer Speech and Language 4 (1), 35–56.

Lari, K., Young, S., 1991. The application of stochastic context-free grammars using the inside-outside algorithm. Computer Speech and Language 5 (3), 237–258.

Levin, E., Pieraccini, R., Jan. 1995. CHRONUS, the next generation. In: Proc. of the DARPA Speech and Natural Language Workshop. Morgan Kaufman Publishers, Inc., Austin, TX, pp. 269–271.

Luo, X., June 2000. Parser adaptation via householder transform. In: Proc. of the IEEE Intl. Conf. on Acoustics, Speech and Signal Processing. Istanbul, Turkey.

Luo, X., Roukos, S., Ward, T., Dec. 1999. Unsupervised adaptation of statistical parsers based on Markov transform. In: IEEE Automatic Speech Recognition and Understanding Workshop. Keystone, Colorado.

Martin, S., Kellner, A., Portele, T., Oct. 2000. Interpolation of stochastic grammar and word bigram models in natural language understanding. In: Proc. of Intl. Conf. on Spoken Language Processing. Beijing, China.

Meng, H., Wai, C., Pieraccini, R., 2000. The use of belief networks for mixed-initiative dialog modelling. In: Proc. of Intl. Conf. on Spoken Language Processing. Beijing, China.

Miller, S., Bates, M., Bobrow, R., Ingria, R., Makhoul, J., Schwartz, R., Jan. 1995. Recent progress in hidden understanding models. In: Proc. of the DARPA Speech and Natural Language Workshop. Morgan Kaufman Publishers, Inc., Austin, TX, pp. 276–280.

Roark, B., Bacchiani, M., May 2003. Supervised and unsupervised PCFG adaptation to novel domains. In: Proceedings of the joint meeting of the North American Chapter of the Association for Computational Linguistics and the Human Language Technology Conference (HLT-NAACL 2003). Edmonton, Canada.

Schabes, Y., 1991. An inside-outside algorithm for estimating the parameters of a hidden stochastic context-free grammar based on earley's algorithm. In: Second Workshop on Mathematics of Language. Tarritown, N.Y.

Schwartz, R., Miller, S., Stallard, D., Makhoul, J., 1997. Hidden understanding models for statistical sentence understanding. In: Proc. of the IEEE Intl. Conf. on Acoustics, Speech and Signal Processing. Munich, pp. 1479–1482.

Seneff, S., 1992. Robust parsing for spoken language systems. In: Proc. of the IEEE Intl. Conf. on Acoustics, Speech and Signal Processing. San Francisco.

Stuttle, M., Williams, J., Young, S., Oct. 2004. A framework for dialog systems data collection using a simulated asr channel. In: Proc. of Intl. Conf. on Spoken Language Processing. Jeju, Korea.

Varga, A., Steeneken, H., Tomlinson, M., Jones, D., 1992. The NOISEX-92 study on the effect of additive noise on automatic speech recognition. Tech. rep., DRA Speech Research Unit.

Ward, W., Issar, S., 1996. Recent improvements in the CMU spoken language understanding system. In: Proc. of the ARPA Human Language Technology Workshop. Morgan Kaufman Publishers, Inc., pp. 213–216.

Williams, J., Young, S., Oct. 2004. Characterising task-oriented dialog using a simulated asr channel. In: Proc. of Intl. Conf. on Spoken Language Processing. Jeju, Korea.

Young, S., 2002. Talking to machines (statistically speaking). In: Proc. of Intl. Conf. on Spoken Language Processing. Denver, Colorado.

Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P., 2004. The HTK Book (for HTK Version 3.2). Cambridge University Engineering Department, http://htk.eng.cam.ac.uk/.