DISCRIMINATIVE SPOKEN LANGUAGE UNDERSTANDING USING WORD CONFUSION NETWORKS

Matthew Henderson, Milica Gašić, Blaise Thomson, Pirros Tsiakoulis, Kai Yu, and Steve Young

Engineering Department, Cambridge University, CB2 1PZ, UK

{mh521, mg436, brmt2, pt344, ky219, sjy}@eng.cam.ac.uk

ABSTRACT

Current commercial dialogue systems typically use hand-crafted grammars for Spoken Language Understanding (SLU) operating on the top one or two hypotheses output by the speech recogniser. These systems are expensive to develop and they suffer from significant degradation in performance when faced with recognition errors. This paper presents a robust method for SLU based on features extracted from the full posterior distribution of recognition hypotheses encoded in the form of word confusion networks. Following [1], the system uses SVM classifiers operating on n-gram features, trained on unaligned input/output pairs. Performance is evaluated on both an off-line corpus and on-line in a live user trial. It is shown that a statistical discriminative approach to SLU operating on the full posterior ASR output distribution can substantially improve performance both in terms of accuracy and overall dialogue reward. Furthermore, additional gains can be obtained by incorporating features from the previous system output.

Index Terms— Semantic decoding, Spoken language understanding, Dialogue systems

1. INTRODUCTION

Spoken Dialogue Systems are often deployed in noisy settings, where background noise, varying line quality and diverse user populations can result in high speech recognition error rates. As a consequence, the semantic representations extracted using hand-crafted semantic decoders such as Phoenix [2] often contain errors. Some mitigation of these errors can be gained by using statistical semantic decoders which have been trained to recognise the correct representation from noisy ASR transcriptions. Examples of such decoders include generative approaches which model the semantics of an utterance as a hidden structure on which observed words are conditioned [3, 4, 5, 6], and discriminative models which train classifiers to directly label utterances using for example conditional random fields [7] and support vector machines (SVMs) [8, 9]. Unlike generative models, discriminative models do not make independence assumptions over the feature set and hence they are generally considered to give better performance [7].

A disadvantage of many discriminative methods however is that they require training utterances to be semantically annotated at the word-level. Manually aligning words with semantic tags is time consuming, and methods which allow training from unaligned data are therefore preferred.

There are a number of possible approaches to training on unaligned data (e.g. [6, 10, 11]). However, a particularly simple but effective approach developed by Mairesse et al. is to view a word string as a collection of n-gram features from which SVM classifiers can be trained to detect tuples from which the required semantics can be reconstructed. In particular, if the semantics denote dialogue acts represented in functor form by a dialogue act type and a list of attribute-value pairs (e.g. `inform(food=indian,area=centre)`) [12], then a multi-class SVM can be used to detect the dialogue act type, and a set of binary SVMs can be used to detect the presence of attribute-value pairs. This *Semantic Tuple Classifier (STC)* approach was shown to significantly outperform a hand-crafted Phoenix parser and was comparable to the best reported results on a benchmark ATIS test set [13]. An extension of this technique forms the basis of this paper.

The impact of speech recognition errors can be further mitigated by using a statistical dialogue manager to track beliefs [14, 15]. The basic idea is that user inputs are treated as evidence from which the user's intentions and beliefs can be inferred. If multiple hypotheses are available, then the dialogue manager can exploit contextual constraints to shape its beliefs and mitigate against errors. For this to work with maximum effect, however, the input evidence at each turn must be a full distribution over all possible semantic interpretations of what the user has just said.

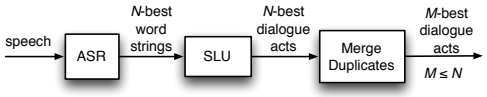
As illustrated in Fig. 1(a), conventional semantic decoders can be extended to provide an approximation of the required distribution by configuring the speech recogniser to generate an N-best list and then applying the decoder to each element of the list in turn. Since similar word strings will frequently map to the same dialogue act, the resulting list must be pruned to remove duplicates. Confidence scores from the recogniser can then be used to weight the resulting M-best list ($M < N$) to form the required distribution. A critical limitation of this approach, however, is that unless very long N-best lists are used, the final pruned lists are often very short[1] and the resulting approximation is quite poor. Secondly each element of the M-best list is being computed from a single element of the N-best word list even though elements of the list are highly correlated. A final limitation is that it is quite difficult in this framework to efficiently apply context constraints such as the previous question to the user.

This paper describes and evaluates an extension to the STC approach in which the decoder is applied directly in a single pass to features extracted from a word confusion network as illustrated in Fig. 1(b). As will be shown by off-line evaluation on a dialogue corpus and by on-line evaluation in a real user trial, this *Confusion network (CNet)* decoder approach outperforms a hand-crafted Phoenix-based decoder and an STC decoder operating in N-best mode. Furthermore it does this in the off-line evaluation for both a 1-best F-measure metric and a full distribution cross-entropy metric, demonstrating that the approach improves both the quality of individual semantic interpretations and the full distribution over all

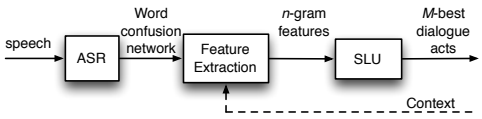
[1] frequently less than 4 for limited domain tasks such as tourist information.

interpretations. Finally, it is shown that the addition of context is both trivial to implement in the CNet framework and effective in further improving performance.

Fig. 1. Semantic decoder configurations: (a) each word string in the N-best list is decoded and any duplicates are merged to give an M-best list of semantic hypotheses; (b) the semantic decoder is applied once to features extracted from a word confusion network and the M-best most likely hypotheses are output. This latter form also facilitates the addition of context by simply extending the feature set.



(a) *N*-best semantic decoding



(b) Confusion network semantic decoding

The remainder of this paper is organised as follows. In section 2, the Phoenix, STC and CNet decoders are described. Section 3 then describes the off-line corpus and the evaluation metrics, and presents the results. Section 4 briefly describes the statistical dialogue system and presents the results of the user trial. In both cases, the application domain is a tourist information service which can locate restaurants according to a variety of criteria such as food type, location, price-range, etc. Finally, section 5 concludes.

2. THE SEMANTIC DECODERS

2.1. Phoenix

Phoenix is a robust semantic decoder which uses manually constructed semantic grammars designed to detect keywords and phrases, and convert them into semantic tags[2]. The Phoenix parser is designed to be robust to word errors and searches to find the best possible parse of the input according to a metric which seeks to find the longest spanning phrases. The grammar used in this evaluation has been specially written for the tourist domain over a period of several years, and has been refined to work effectively on the ASR hypotheses generated by our recognition system[2].

The Phoenix decoder is configured as in Fig. 1(a) to run on the top 10 ASR hypotheses, producing a single dialogue act for each. These are then weighted by the corresponding posterior probability assigned by the recogniser to each hypothesis in the N-best list, and duplicates are merged to give the final distribution over dialogue acts.

2.2. Semantic Tuple Classifier

As explained in the introduction, the STC decoder detects dialogue act types and slot attribute-value pairs based on the N-gram counts in the input word sequence [1]. The STC decoder requires a set of SVMs to be trained: one multi-class classifier is used to predict the dialogue act type, and a binary classifier is used to predict the existence of each possible slot-value pair. The SVM classifiers use a linear kernel, and outputs are converted to probabilities using a sigmoid function [17].

Running the classifiers on an utterance, u, gives the probability of each dialogue act type, $P(\text{d-type}_i \mid u)$ as well as the probability of each possible slot-value pair, $P(sv|u)$. The probability of a dialogue-act D of type d-type$_j$ with a set of slot-value pairs, S is approximated by:

$$P(D \mid u) = P(\text{d-type}_j|u) \prod_{sv \in S} P(sv|u) \prod_{sv \notin S} (1 - P(sv|u)) \quad (1)$$

A simple set of rules is imposed on S for each dialogue act type, to ensure that the resulting dialogue act makes sense:

- S must be empty for dialogue act types: ack, bye, null, repeat, restart, thankyou
- S must contain at least one unbounded slot if and only if the dialogue act type is request
- S must contain one bounded slot for types: confirm, inform, reqalts

where a bounded slot is one which occurs in a pair with a specified value, as in 'food=chinese' and an unbounded slot does not, e.g. 'phone' which signifies it is being requested. Subject to these rules, a search is performed to find the top M most probable dialogue acts.

The vector representation of an utterance, u, as presented in [1] is $x_i = C_u(n\text{-gram}_i)$ where $C_u(ng)$ counts how many times the n-gram ng occurs in utterance u. n is allowed to range from 1 to 3, i.e. words and sequences of two and three words are counted. Only a small number of n-grams present in the training data occur in any single utterance, meaning this is a sparse representation. This allows for fast training and classification with SVMs.

Mairesse et al. evaluated the performance of the STC model trained on only the top ASR hypothesis and such a system was shown to outperform the Phoenix baseline [1]. As a better comparison for the CNet decoder described in the next section, it is possible to extend the training of the STC decoder to use the hypotheses in the ASR N-best lists. Two methods of doing this have been explored.

n-gram Features

The obvious way to exploit the N-best lists for training is to create a data point for each of the top k hypotheses with the same semantic reference label (dialogue act type or slot-value pair). To take account of the confidence scores attached to the hypotheses, each data point in the SVM training algorithm is assigned a misclassification cost which is proportional to the posterior probability of the hypothesis. A major limitation of this approach, however, is that the size of the training set is multiplied by k and since the training time of an SVM depends on the training set size R roughly as $\mathcal{O}\left(R^3\right)$, k must be kept very small.

Weighted Sum Features

To avoid increasing the training set size, the n-gram features from each of the top N ASR hypotheses can be weighted by their posterior probability and then summed. This attempts to summarise the information in the N best list, and does not have the problem of increasing the training set size. This representation can be written as:

$$x_i = \sum_{j=1}^{N} C_{hyp_j}(n\text{-gram}_i) \cdot p_j$$

[2] A HTK-based system with a vocabulary of around 3000 words and a trigram language model [16]

where x_i is the i'th element of the training vector and p_j the posterior probability of hyp_j, the j'th ASR hypothesis in the N-best list.

2.3. Confusion Network Decoder

A word confusion network is an efficient structure for representing the lattice of all hypotheses generated by a speech recogniser [18]. Confusion network features are shown in [19] to be useful in developing more robust systems for named entity extraction and call classification. They allow for efficient calculation of the quantities $\mathbb{E}\left(C_u(n\text{-gram}_i)\right)$, the expected frequency of n-gram$_i$ in the utterance (if the n-gram only appears in one path in the graph, then this is just its probability of occurrence.)

The CNet decoder is trained and configured in a similar way to the STC decoder above, the only difference being that the recogniser is configured to output confusion networks rather than N-best lists and the SVMs are trained from a single feature vector with elements:

$$x_i = \mathbb{E}\left(C_u(n\text{-gram}_i)\right)^{1/|n\text{-gram}_i|}$$

where $|n\text{-gram}_i|$ is the number of words in n-gram$_i$. The exponentiation is a normalisation included to compensate for the fact that longer word sequences are always less likely than their subsequences.

At run-time, the confusion network decoder is applied just once to each ASR output and the top M hypotheses are generated in rank order according to equation (1).

2.4. Dialogue Context Features

To investigate the effectiveness using the dialogue context to constrain the semantic decoder, a set of context features y_i are extracted from the last system act as $y_i = \iota(x_i \in D_m)$, where ι is an indicator function and x_i runs over all dialogue act types and slot-value pairs to find occurrences in the last system dialogue act D_m[3]. The final representation of a user utterance when using this context is then a concatenation of the original word confusion network features and the context features. This allows the models to learn a dependence on the last act which the machine generated.

Incorporating this level of context does not lead to an overly restrictive decoder. This was confirmed by comparing the performance of the learnt models when faced with instances where the system requested information from the user, but the user did not provide it. Two decoders using confusion network features were evaluated on the subset of such instances, one of which used context features and the other did not. The difference in performance was negligible suggesting that the models learned to not rely too heavily on context from the examples in the training corpus.

3. OFF-LINE CORPUS EVALUATION

3.1. Cambridge Restaurant Information Domain

The dialogue corpus data used for off-line evaluation of the decoders described above was collected using a restaurant information system for the City of Cambridge. Users can specify restaurant suggestions by area, price-range and food type and and can then query the system for additional restaurant specific information such as phone number, post code, signature dish and address. To achieve a range of differing noise conditions, participants were asked to interact with different systems operating in a variety of conditions related to in-car dialogues; in a stationary car with the air conditioning fan on and off, in a moving car and in a car simulator [20, 21]. For the creation of this corpus, all of the utterances from the dialogues were re-transcribed using the same speech recogniser, which achieved an average word error rate (WER) of 37.0%.

Each section of the corpus has an equal distribution of the different in-car conditions. Some basic statistics of the corpora are given in Table 1, and the data is available for download online[4].

	Training	**Testing**
Dialogues	1522	644
Utterances	10571	4882
Male : Female	28 : 31	15 : 15
Native : Non-Native	33 : 26	21 : 9

Table 1. Training and Testing Corpora

3.2. Dialogue Acts

The outputs of the semantic decoders are dialogue acts, conforming to the Cambridge Dialogue Act Specification [12]. A dialogue act is specified by a *dialogue act type* which describes the type of action the user is trying to perform, and a set of slot-value pairs which specify what constraints are being imposed. If a slot is being requested, then its value is simply omitted. For clarification, see Table 2 which has a list of examples of dialogue acts in this domain.

Transcription	**Dialogue Act**
I want an Indian restaurant in the west of town.	inform {food=indian, area=west, type=restaurant}
Good bye.	bye {}
Do you know the phone number of one in the east?	request {phone, area=east}
Is that in the east?	confirm {area=east}
Any part of town.	inform {area=dontcare}
Is there anything else?	reqalts {}
No I want a cheap Chinese place.	negate { pricerange=cheap, food=chinese }
I don't want British.	deny { food=british }

Table 2. Example dialogue acts, consisting of a dialogue act type and a set of slot-value pairs. Requested slots do not have values, for example 'phone' in the third example.

3.3. Evaluation Metrics

For conventional, non-statistical dialogue systems, an important metric to assess the quality of the output of a semantic decoder is the **F-score**, the harmonic mean of the precision and recall of true semantic items in the top semantic hypothesis.

Let the true reference dialogue act for an utterance be D_{ref}, given by the dialogue act type d-type$_{\text{ref}}$ and the set of slot-value pairs S_{ref}. Suppose that a semantic decoder output the hypotheses D_{hyp_i} with

[3] User and system dialogue acts have exactly the same form in the CUED system

[4] http://mi.eng.cam.ac.uk/∼mh521/incarslu

corresponding probabilities p_i for $i = 0, \ldots, m-1$ and $p_0 \geq p_1 \geq p_2 \geq \ldots \geq p_{m-1}$, where D_{hyp_i} is given by the dialogue act type d-type$_{\text{hyp}_i}$ and the set of slot-value pairs S_{hyp_i}, then the F-score of this decoding output is:

$$\begin{aligned} F &= \tfrac{2|A \cap B|}{|A|+|B|} \\ \text{where } A &= (S_{\text{ref}} \cup \{\text{d-type}_{\text{ref}}\}) \\ \text{and } B &= (S_{\text{hyp}_0} \cup \{\text{d-type}_{\text{hyp}_0}\}) \end{aligned}$$

As noted in the introduction, in a statistical dialogue system, the distribution over all possible dialogue acts is used to update the dialogue belief state. It is therefore important that the distribution output by the semantic decoder accurately reflects the underlying uncertainty. The Item Cross Entropy (**ICE**) between the hypotheses and the true semantics measures the overall quality of a distribution and is shown to provide consistent rankings between semantic decoders [22].

The ICE score is calculated from the confidences, $c(\cdot)$, of each dialogue act type and slot-value pair:

$$\begin{aligned} c(\text{d-type}) &= \sum_{i=0}^{M-1} \begin{cases} p_i & \text{if d-type} = \text{d-type}_{\text{hyp}_i} \\ 0 & \text{otherwise} \end{cases} \\ c(sv) &= \sum_{i=0}^{M-1} \begin{cases} p_i & \text{if } sv \in S_{\text{hyp}_i} \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

These are compared with the true distribution of the semantics, $c^\star(\cdot)$,

$$\begin{aligned} c^\star(\text{d-type}) &= \begin{cases} 1 & \text{if d-type} = \text{d-type}_{\text{ref}} \\ 0 & \text{otherwise} \end{cases} \\ c^\star(sv) &= \begin{cases} 1 & \text{if } sv \in S_{\text{ref}} \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Giving the cross entropy, or ICE:

$$\text{ICE} = \frac{1}{1+|S_{\text{ref}}|} \sum_{x \in X} \log\left(c(x)c^\star(x) + (1-c(x))(1-c^\star(x))\right)$$

where X is a set containing all the possible dialogue act types and slot value pairs, and the arguments to the log function are thresholded to prevent attempting to calculate $\log(0)$.

The final metric reported is the dialogue act type accuracy (**DA type Acc.**), which measures whether the top semantic hypothesis has the correct dialogue act type. In the notation above:

$$\text{DA type Acc.} = \begin{cases} 1 & \text{if d-type}_{\text{hyp}_0} = \text{d-type}_{\text{ref}} \\ 0 & \text{otherwise} \end{cases}$$

In the evaluation, the training corpus is used to train a semantic decoder, then the whole test corpus is decoded. The F-score, ICE and dialogue act type accuracy for each test utterance are then calculated and averaged.

3.4. Experimental Results

Table 3 shows the performance of various semantic decoders in terms of the F-score, Item Cross Correlation (ICE) and dialogue act type accuracy achieved in the test corpus.

The n-gram features are evaluated with k (the number of ASR hypotheses used for training) set at 1 and 2. Training complexity becomes an issue for higher values, and the increase in performance is negligible. The pay-off for increasing k was found to be higher for smaller training corpora. The systems trained on these features

Features	**Trained on**	**Context Features**	**F Score**	**ICE**	**DA type Acc.**	
Phoenix	(Hand-crafted grammar)	no	0.694 ± 0.012	2.784 ± 0.116	0.706 ± 0.013	1
n-grams from N-best list hypotheses	top hypothesis	no	0.692 ± 0.012	1.790 ± 0.065	0.706 ± 0.013	2
	top 2 hypotheses		0.703 ± 0.012	1.719 ± 0.068	0.724 ± 0.013	3
	top hypothesis	yes	0.725 ± 0.011	1.529 ± 0.062	0.754 ± 0.012	4
	top 2 hypotheses		0.740 ± 0.011	1.499 ± 0.064	0.773 ± 0.012	5
Weighted sum of vectors from N-best list	$N = 10$	no	0.708 ± 0.012	1.760 ± 0.074	0.729 ± 0.012	6
		yes	0.742 ± 0.011	1.497 ± 0.066	0.773 ± 0.012	7
Confusion network features	Full confusion network	no	0.730 ± 0.011	1.680 ± 0.062	0.757 ± 0.012	8
		yes	**0.767 ± 0.011**	**1.431 ± 0.063**	**0.800 ± 0.011**	9

Table 3. Results are written $\mu \pm 1.96\sigma$ where μ is the estimate of the mean over the utterances in the test corpus and σ the standard error. Row 2 corresponds to the basic STC decoder described in [1].

(rows 2 and 3) achieve F-scores comparable to Phoenix, and significantly smaller (i.e. better) ICE scores. The decrease in ICE score and improved dialogue act type accuracy (DA type Acc.) resulting from increasing k from 1 to 2 is probably significant. Incorporating the last system act context features (rows 4 and 5) increases the F-scores to be higher than the Phoenix result.

The results of the decoders trained on weighted sum features (rows 6 and 7) are similar to the $k = 2$ system (rows 3 and 5), suggesting this is a reasonable method of summarising the information in the N-best list. Recall this representation avoids the problem of multiplying the size of the training set.

The decoders using confusion network features (rows 8, 9) perform well compared to the others. The results suggest that the F-score of the context independent decoder (row 8) is better than that of any other context-dependent decoder. The context dependent decoder (row 9) scored better than any other system in F-score, ICE and DA type Acc. The confusion network features can be thought of as similar to weighted sum features in the limit of increasing k, the number of top ASR hypotheses used. Intuitively, there is more information in these features, as they may pick out n-grams which do not appear in the top N hypotheses and furthermore assign them weights which more accurately reflect our estimate of the expected n-gram counts.

Figure 2 shows an example where the keyword 'west' is not in the top 10 ASR hypotheses, causing the hand-crafted grammar to

fail. The word 'west' is found (although with a low weight) in the confusion network, and the statistical models have learnt typical confusions of the speech recogniser, allowing it to give some weight to the correct hypothesis 'inform(area=west)'. Because the last system act in the dialogue was asking the user to select between the west area and any area, the model with context puts an even higher weight on the correct hypothesis.

To show that the best system, confusion network features with context, is more robust to noise than the Phoenix baseline, polynomial regressions were run for the two systems predicting ICE and F-score from the utterance Word Error Rate. For the F-score, a degree 2 polynomial was found to model the data best, and for ICE a linear regression was found to be best using F-tests. Figure 3 shows the results of these regressions.

The regressions suggest that the statistical decoder has learnt a decoder which degrades much more gracefully when faced with speech recognition errors than the hand-crafted grammar.

4. USER TRIAL

A user trial was run to investigate the effect of using the best statistical decoder found in the previous section, the confusion network and context features decoder (Table 3, row 9), in the context of an end-to-end dialogue system. The hand-crafted Phoenix grammar (Table 3, row 1) was used to provide a baseline.

4.1. Experimental Set-up

One hundred native speakers of American English were recruited using Amazon Mechanical Turk [23]. Each was asked to use the dialogue system to find a restaurant in Cambridge matching a set of constraints, and to then request some details. Some tasks involved specifying constraints which should be relaxed in case no matching venue was found. An example of a typical task was: 'Try to find a Chinese restaurant in the west, if there is none then try Thai food. Get the phone number and address.' After a dialogue, the participant was asked whether or not they got the information they needed, and if they agreed then the dialogue was recorded as being successful.

Participants were randomly allocated either a system using the Phoenix grammar, or one using the Confusion network with context decoder. Both dialogue systems use the Bayesian Update of Dialogue State framework to track the dialogue belief state, treating the dialogue planning process as a Partially Observable Markov Decision Process (POMDP) [15]. The dialogue policy for each system

Fig. 2. Illustrative example

Last system act: select(area=west,area=dontcare)
Transcription: west side of town please
ASR: what kind of town please, what what kind of town please, kind of town please, etc.

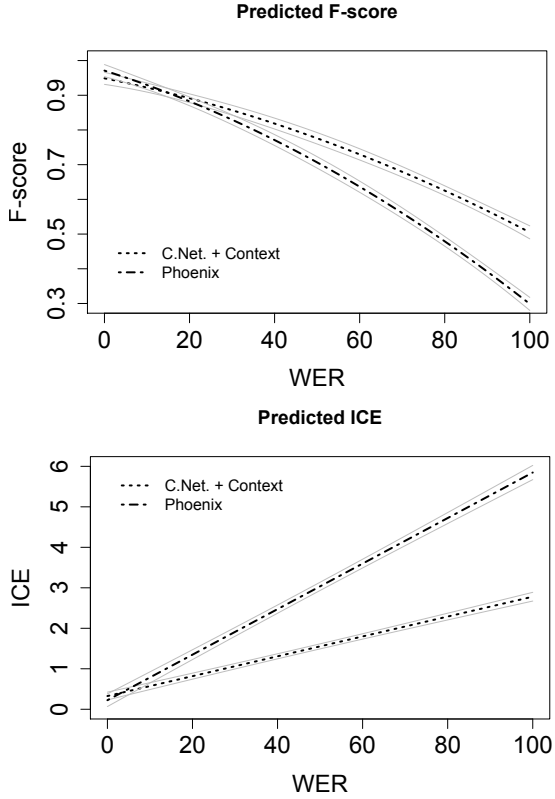
M-best dialogue acts:

CNet decoder, with context:
0.79 inform(area=west)
0.15 inform(area=north)
0.05 request(food, area=west)
0.01 null()

CNet decoder, no context:
0.37 null()
0.32 inform(area=west)
0.14 inform(area=north)
0.09 reqalts()
0.07 request(food)
0.01 inform(area=centre)

Phoenix grammar:
0.96 null()
0.04 request()

Fig. 3. Regressions of F-score and ICE against WER. Grey lines show margins of 2 standard errors. Confusion network decoder is shown to degrade significantly more gracefully as noise increases.



was optimised to maximise a reward function R using the Natural Actor Critic learning algorithm where

$$R = 20 \cdot \text{Success} - \text{Number of user turns} \quad (2)$$

and Success = 1 if the dialogue is successful, and 0 otherwise. This reward function provides a measure of dialogue quality reflecting the design objective of achieving a high success rate and short dialogues. Each policy was trained using a simulated user with built-in error model and each error model was separately trained to reflect the type of confusions each decoder makes, using real example confusions.

Note that unlike the off-line corpus, the live trial was conducted using relatively clean telephone calls and a different set of acoustic models optimised for this domain. The WER over the 924 trial dialogues was 20.1%, while the WER in the off-line corpus was 37.0%.

4.2. Trial Results

The results of the trial are shown in Table 4. The F-scores, ICE scores and Dialogue Reward achieved by the CNet decoder are significantly better than the Phoenix grammar. The raw success rate is also higher for the CNet decoder although the difference is not statistically significant. The average dialogue length is shorter by half a turn on average and this is significant. Overall the differences between the two decoders are not as pronounced as in the offline evaluation because the dialogues in the trial were at much lower word error rates. The high success rates achieved in the User Trial are indicative of this. Given the evidence of the off-line evaluation, it is hypothesised that the advantages of the CNet decoder would become more pronounced in more challenging scenarios such as in a car using an open far-field microphone.

	Phoenix	CNet with context
Dialogues	456	468
F-score	0.795 ± 0.02	0.822 ± 0.02
ICE	2.016 ± 0.223	1.264 ± 0.151
Success Rate (%)	94.3 ± 2.0	94.7 ± 2.0
Turns per dialogue	7.25 ± 0.29	6.79 ± 0.25
Dialogue Reward	9.60 ± 0.55	10.15 ± 0.54

Table 4. Results of User Trial. Errors are 1.96 times the standard error.

5. CONCLUSIONS

Building on the Semantic Tuple Classifier (STC) proposed by Mairesse et al., this paper has described a statistical Confusion Network (CNet) semantic decoder which is applied directly in a single pass to features extracted from a word confusion network. It has been shown through off-line evaluation on a dialogue corpus collected in noisy conditions that the CNet decoder approach outperforms both a hand-crafted Phoenix-based decoder and an STC decoder operating in N-best mode. Furthermore it does this for both a 1-best F-measure metric and a full distribution cross-entropy metric, demonstrating that the approach improves both the quality of individual semantic interpretations and the full distribution over all interpretations. It has also been shown that the addition of context is both simple to implement in the CNet framework and effective in further improving performance. Finally, it has been shown via a user trial that the performance advantages indicated by off-line evaluation do translate into improved overall performance when used in a live dialogue system

6. ACKNOWLEDGEMENTS

The principal author was funded by a studentship from the EPSRC and the work was supported in part by PARLANCE (www.parlance-project.eu), an EU Seventh Framework Programme project (grant number 287615).

7. REFERENCES

[1] F. Mairesse, M. Gašić, F. Jurcicek, S. Keizer, B. Thomson, K. Yu, and S. J. Young, "Spoken language understanding from unaligned data using discriminative classification models," in *Proceedings of ICASSP*, 2009.

[2] W. Ward, "Extracting Information From Spontaneous Speech," in *Proceedings of INTERSPEECH*, 1994.

[3] R. Schwartz, S. Miller, D. Stallard, and J. Makhoul, "Language understanding using hidden understanding models," in *ICSLP*, 1996.

[4] E. Levin and R. Pieraccini, "Chronus, the next generation," in *Proceedings of the ARPA Workshop on Spoken Language Technology*, 1995.

[5] A. Acero and Y. Y. Wang, "Spoken language understanding," *IEEE Signal Processing Magazine*, vol. 22, no. 5, Sept. 2005.

[6] Y. He and S. J. Young, "Spoken language understanding using the Hidden Vector State Model," *Speech Communication*, vol. 48, no. 3-4, pp. 262–275, Mar. 2006.

[7] Y. Y. Wang and A. Acero, "Discriminative models for spoken language understanding," in *INTERSPEECH*, 2006.

[8] S. Pradhan, W. Ward, K. Hacioglu, J. Martin, and D. Jurafsky, "Shallow semantic parsing using support vector machines," in *Proceedings of HLT/NAACL*, 2004, p. 233.

[9] R. J. Kate and R. J. Mooney, "Using string-kernels for learning semantic parsers," in *Proceedings of ACL*, 2006.

[10] I. V. Meza-Ruiz, S. Riedel, and O. Lemon, "Accurate statistical spoken language understanding from limited development resources," in *Proceedings of ICASSP*, 2008.

[11] D. Zhou and Y. He, "Learning conditional random fields from unaligned data for natural language understanding," in *Advances in Information Retrieval*, Lecture Notes in Computer Science, pp. 283–288. Springer Berlin / Heidelberg, 2011.

[12] S. J. Young, "CUED standard dialogue acts," *Report, Cambridge University Engineering Department, 14th October*, 2007.

[13] L.S. Zettlemoyer and M. Collins, "Online learning of relaxed CCG grammars for parsing to logical form," in *Proceedings of EMNLP-CoNLL*, 2007.

[14] J. D. Williams and S. J. Young, "Partially Observable Markov Decision Processes for Spoken Dialog Systems," *Computer Speech and Language*, vol. 21, no. 2, pp. 393–422, 2007.

[15] B. Thomson and S. J. Young, "Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems," *Computer Speech & Language*, vol. 24, no. 4, Oct. 2010.

[16] S. J. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book, version 3.4*, Cambridge University Engineering Department, Cambridge, UK, 2006.

[17] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[18] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus among words: lattice-based word error minimisation," *Computer Speech & Language*, pp. 373–400, 2000.

[19] D. Hakkani-Tür, F. Béchet, G. Riccardi, and G. Tur, "Beyond ASR 1-best: Using word confusion networks in spoken language understanding," *Computer Speech & Language*, vol. 20, no. 4, pp. 495–514, Oct. 2006.

[20] M. Gašić, P. Tsiakoulis, M. Henderson, B. Thomson, K. Yu, E. Tzirkel, and S. J. Young, "The effect of cognitive load on a statistical dialogue system," in *Proceedings of SIGdial*, 2012.

[21] P. Tsiakoulis, M. Gašić, M. Henderson, J. Prombonas, B. Thomson, K. Yu, S. J. Young, and E. Tzirkel, "Statistical methods for building robust spoken dialogue systems in an automobile," in *4th International Conference on Applied Human Factors and Ergonomics*, 2012.

[22] B. Thomson, K. Yu, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, and S. J. Young, "Evaluating semantic-level confidence scores with multiple hypotheses," in *INTERSPEECH*, 2008.

[23] F. Jurcicek, S. Keizer, M. Gašić, F. Mairesse, B. Thomson, K. Yu, and S. J. Young, "Real user evaluation of spoken dialogue systems using Amazon Mechanical Turk," in *Proceedings of INTERSPEECH*, 2011, pp. 3061–3064.