Exploiting Sentence and Context Representations in Deep Neural Models for Spoken Language Understanding

Lina M. Rojas-Barahona, Milica Gašić, Nikola Mrkšić, Pei-Hao Su
Stefan Ultes, Tsung-Hsien Wen and Steve Young
Department of Engineering, University of Cambridge, Cambridge, UK
lmr46, mg436, nm480, phs26, su259, thw28, sjy@cam.ac.uk

Abstract

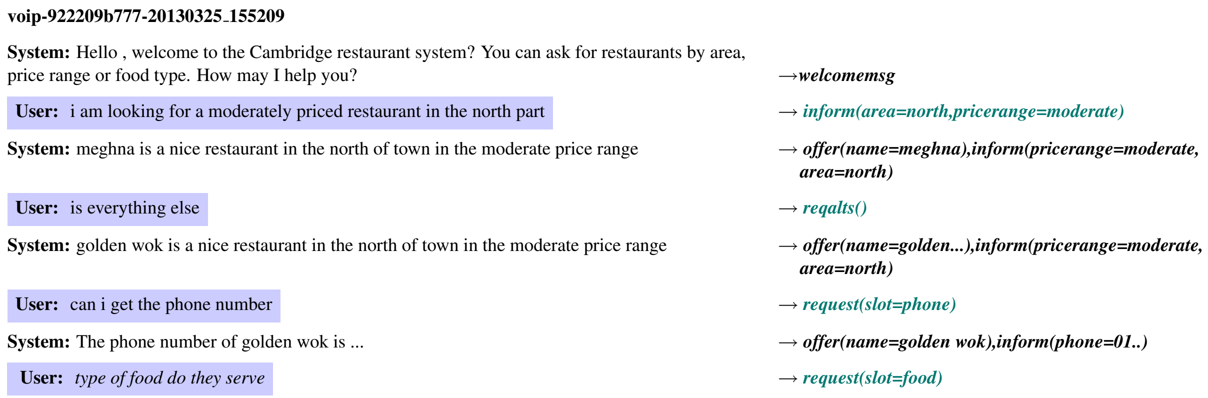
This paper presents a deep learning architecture for the semantic decoder component of a Statistical Spoken Dialogue System. In a slot-filling dialogue, the semantic decoder predicts the dialogue act and a set of slot-value pairs from a set of n-best hypotheses returned by the Automatic Speech Recognition. Most current models for spoken language understanding assume (i) word-aligned semantic annotations as in sequence taggers and (ii) delexicalisation, or a mapping of input words to domain-specific concepts using heuristics that try to capture morphological variation but that do not scale to other domains nor to language variation (e.g., morphology, synonyms, paraphrasing). In this work the semantic decoder is trained using unaligned semantic annotations and it uses distributed semantic representation learning to overcome the limitations of explicit delexicalisation. The proposed architecture uses a convolutional neural network for the *sentence representation* and a long-short term memory network for the *context representation*. Results are presented for the publicly available DSTC2 corpus and an In-car corpus which is similar to DSTC2 but has a significantly higher word error rate (WER).

1 Introduction

In most existing work on Spoken Language Understanding (SLU), semantic decoding is usually seen as a sequence tagging problem with models trained and tested on datasets with word-level annotations (Tür et al., 2013; Mesnil et al., 2015; Yao et al., 2013; Sarikaya et al., 2011; Deoras and Sarikaya, 2013; Sarikaya et al., 2014). Spoken language understanding from *unaligned data*, in which utterances are annotated with an abstract semantics, faces the additional challenge of not knowing which specific words are relevant for extracting the semantics. This problem was tackled in (Zhou and He, 2011), by using conditional random fields (CRFs) driven by finely-tuned hand-crafted features. Other discriminative approaches that deal with unaligned data use some form of *delexicalisation* or mapping of the input to known ontological concepts (Henderson et al., 2012; Henderson et al., 2014a). The main disadvantage of delexicalisation is the difficulty in scaling it, not only to larger and more complex dialogue domains but also to handle the many forms of language variation.

We propose in this paper a semantic decoder that learns from unaligned data (Figure 1) and that exploits rich semantic distributed word representations instead of delexicalisation. The semantic decoder predicts the dialogue act and the set of slot-value pairs from a set of n-best hypotheses returned by an automatic speech recognition (ASR). The prediction is made in two steps. First, a deep learning architecture is used for the joint prediction of dialogue acts and the presence or absence of slots. Second, the same architecture is reused for predicting the values of the slots that were detected by the first joint-classifier. The deep architecture combines sentence and context representations. A convolutional neural network (CNN) (Collobert et al., 2011) is used to generate the sentence representation, while a long-short term memory (LSTM) network (Hochreiter and Schmidhuber, 1997) is used to generate the context representation. A non-linear function then combines the top layers of these neural networks and distinct softmax layers are used to predict the dialogue act and slots in the first joint model. In the second model, a single softmax predicts the possible values for each slot.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 258–267, Osaka, Japan, December 11-17 2016.



voip-922209b777-20130325_155209

System: Hello , welcome to the Cambridge restaurant system? You can ask for restaurants by area, price range or food type. How may I help you?	→*welcomemsg*
User: i am looking for a moderately priced restaurant in the north part	→ *inform(area=north,pricerange=moderate)*
System: meghna is a nice restaurant in the north of town in the moderate price range	→ *offer(name=meghna),inform(pricerange=moderate, area=north)*
User: is everything else	→ *reqalts()*
System: golden wok is a nice restaurant in the north of town in the moderate price range	→ *offer(name=golden...),inform(pricerange=moderate, area=north)*
User: can i get the phone number	→ *request(slot=phone)*
System: The phone number of golden wok is ...	→ *offer(name=golden wok),inform(phone=01..)*
User: *type of food do they serve*	→ *request(slot=food)*

Figure 1: Excerpt from a dialogue in the DSTC2 corpus. The top-best ASR hypothesis is shown highlighted on the left, and the corresponding user semantics is shown highlighted on the right.

Our models are evaluated on two datasets DSTC2 (Henderson et al., 2014b) and In-car (Tsiakoulis et al., 2012) using accuracy, f-measure and the Item Cross Entropy (ICE) score (Thomson et al., 2008). We show that these models outperform previous proposed models, without using manually designed features and without any preprocessing of the input (e.g., stop words filtering, delexicalisation). They do this by exploiting distributed word representations and we claim that this allows semantic decoders to be built that can easily scale to larger and more complex dialogue domains.

The remainder of this paper is structured as follows. We first present related work in Section 2 and then we describe our architecture in Section 3. We describe the experimental setup in 4 and the evaluation results are introduced in Section 5. Finally, we present conclusions and future work in Section 6.

2 Related Work

Sequence tagging discriminative models such as CRFs and sequence neural networks have been widely explored for spoken language understanding. For instance, Recurrent Neural Networks have been proposed in (Yao et al., 2013; Mesnil et al., 2015) and generative Deep Neural Networks consisting of a composition of Restricted Boltzmann Machines (RBM) have been studied by (Sarikaya et al., 2011; Deoras and Sarikaya, 2013; Sarikaya et al., 2014). A combination of neural networks and triangular CRFs is presented in (Celikyilmaz and Hakkani-Tur, 2010), in which a convolutional neural network is used for extracting the input features of a triangular CRF in order to perform joint intent detection and slot filling. All these models use word-level semantic annotations. However, providing these word-level semantic annotations is costly since it requires specialised annotators. (Zhou and He, 2011) has proposed learning CRFs from unaligned data, however they use manually tuned lexical or syntactic features. In this work we avoid the need for word-level annotation by exploiting distributed word embeddings and using deep learning for feature representation.

Convolutional Neural Networks (CNNs) have been used previously for sentiment analysis (Kim, 2014; Kalchbrenner et al., 2014) and in this work we explore a similar CNN to the one presented by Kim (2014) for generating a sentence representation. However unlike Kim (2014), the input in not a single well formed sentence but a set of ill-formed ASR hypotheses. Additionally, the softmax layer used for binary classification (i.e., positive or negative sentiment) is replaced by a softmax layer for multiclass dialogue act prediction and a further softmax layer is added for each distinct slot in the domain. (Chen and He, 2015) proposed a CNN for generating intent embeddings in SLU, which uses tri-letter input vectors. Instead, in this paper the models are initialised with GloVe word embeddings (Pennington et al., 2014). These GloVe embeddings were trained in an unsupervised fashion on a large amount of data to model the contextual similarity and correlation between words. Chen and He's model aims to learn the embeddings for utterances and intents such that utterances with similar intents are close to each other in the continuous space. Although we share the same spirit, we use sentence embeddings not only for intent (or dialogue act) recognition but also for slot-filling within a dialogue system and we combine them with embeddings for dialogue context.

Approaches for adaptive SLU have been proposed in (Ferreira et al., 2015; Zhu et al., 2014), however they focused more on domain adaptation on top of an existing SLU component. Moreover, they use classical discriminative models for SLU such as CRFs and SVMs that require manually designed features. In contrast, the focus of this paper is to exploit deep learning models for SLU, which learn feature representations automatically.

Recently, some researchers have focused on mapping word level hypotheses directly to beliefs without using an explicit semantic decoder step (Henderson et al., 2014a; Mrkšić et al., 2015). These systems track the user's goal through the course of the dialogue by maintaining a distribution over slot-value pairs. Such systems are interesting, but it is not clear that they can be scaled to very large domains due to the constraint of delexicalisation. Furthermore, they still require an explicit semantic decoding layer for domain identification and general *Topic Management*.

3 Deep Learning Semantic Decoder

We split the task of semantic decoding into two steps: (i) training a joint model for predicting the dialogue act and presence or absence of slots and (ii) predicting the values for the most probable slots detected in (i). As shown in Figure 2, we use the same deep learning architecture in both steps for combining sentence and context representations to generate the final hidden unit that feeds one or many softmax layers. In the first step, as shown in the Figure, there are distinct softmax layers for the joint optimisation of the dialogue act and each possible slot. In the second step there is a single softmax layer that predicts the value of each specific slot. In the following we explain this architecture in more detail.

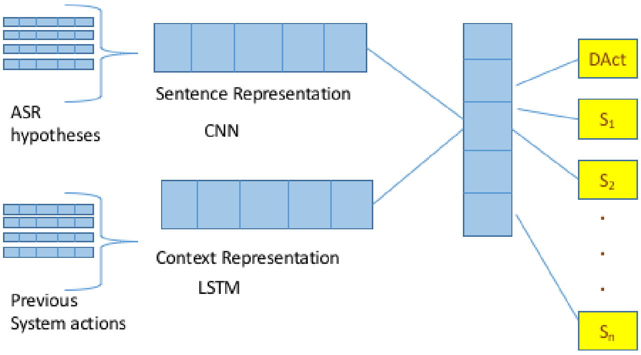


Figure 2: Combination of sentence and context representations for the joint prediction of dialogue acts and slots.

3.1 Sentence Representation

A CNN is used for generating the hypothesis representation, then these representations are weighted by their confidence scores and then summed up to obtain the sentence representation (Figure 3).

The CNN is a variant of (Kim, 2014), in which the inputs are the word vectors in each *ASR hypothesis*. Let x_i be a $k-$dimensional word embedding for the i-th word in a hypothesis. A hypothesis of length m is represented as: $x_{1:m} = \mathbf{x}_1 \bigoplus \mathbf{x}_2 \bigoplus ... \bigoplus \mathbf{x}_m$ where $\bigoplus$ is the concatenation operator. A convolutional operation is applied to a window of l words to produce a new feature.

$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+l-1} + b) \tag{1}$$

where f is the hyperbolic tangent function; $w \in \mathbb{R}^{lk}$ is a filter applied to a window of l words and $b \in \mathbb{R}$ is a bias term. The filter is applied to every window of words in the sentence to produce a feature map.

$$\mathbf{c} = [c_1, c_2, ..., c_{n-l+1}] \tag{2}$$

with $\mathbf{c} \in \mathbb{R}^{n-l+1}$. A max pooling operation is then applied to give the maximum value $c = max\{\mathbf{c}\}$ as the representative feature for that filter. Multiple filters can be applied by varying the window size to obtain several adjacent features for a given hypothesis. These features $\hat{f}_j$ for the hypothesis $j \in H$ are then multiplied by the ASR confidence score p_j[1] and summed over all ASR hypotheses to generate a representation for the sentence s_t (Equation 3), as shown in Figure 3.

$$s_t = \sum_{j \in H} \hat{f}_j * p_j \tag{3}$$

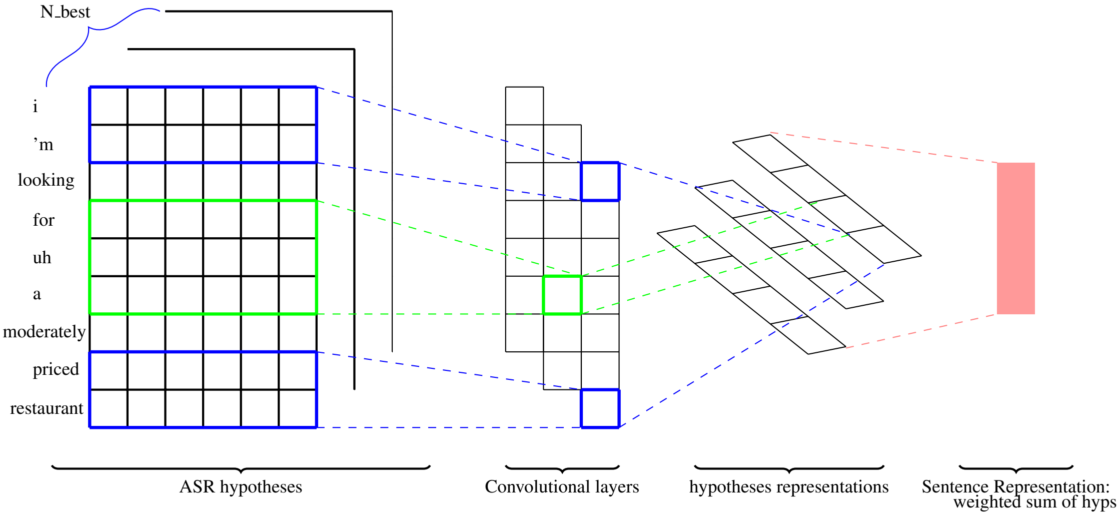


Figure 3: Sentence Representation: after applying convolution operations on the N-best list of ASR hypotheses, the resulting hidden layers are weighted by the ASR confidence scores and summed.

3.2 Context Representation

An LSTM (Hochreiter and Schmidhuber, 1997) is used for tracking the context implied by previous dialogue system actions. The top layer of this LSTM network then provides the context representation for decoding the current input utterance.

An LSTM is a sequence model that utilises a memory cell capable of preserving states over long periods of time. This cell is recurrently connected to itself and it has three multiplication units, an input gate, a forget gate and an output gate. These gating vectors are in [0,1]. The cell makes selective decisions about what information is preserved, and when to allow access to units, via gates that open and close. The LSTM transition equations are as follows:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}^{(i)} \cdot x_t + \mathbf{U}^{(i)} \cdot h_{t-1} + b^{(i)}), \\
\mathbf{f}_t &= \sigma(\mathbf{W}^{(f)} \cdot x_t + \mathbf{U}^{(f)} \cdot h_{t-1} + b^{(f)}), \\
\mathbf{o}_t &= \sigma(\mathbf{W}^{(o)} \cdot x_t + \mathbf{U}^{(o)} \cdot h_{t-1} + b^{(o)}), \\
\mathbf{u}_t &= tanh(\mathbf{W}^{(u)} \cdot x_t + \mathbf{U}^{(u)} \cdot h_{t-1} + b^{(u)}), \\
\mathbf{c}_t &= \mathbf{i}_t \bigodot \mathbf{u}_t + \mathbf{f}_t \bigodot \mathbf{c}_{t-1}, \\
\mathbf{h}_t &= \mathbf{o}_t \bigodot tanh(\mathbf{c}_t)
\end{aligned} \tag{4}$$

where h_t is the hidden unit at time step t, x_t is the input at the current time step, b is a bias, σ is the logistic sigmoid function and $\bigodot$ denotes elementwise multiplication.

As shown in Figure 1, system actions are encoded in the form of a system dialogue act plus one or more slot-value pairs. To track the history of system actions, slots and values are treated as words and the input x_t is formed from its corresponding word vectors. The length of the context can vary. We consider

[1] The posterior probability of hypothesis j in the N-best list.

all the system actions previous to the current user utterance, or a window l of the previous system actions. For instance, if we are currently processing the last user input in Figure 1, in which L is the total number of system actions, we can consider all previous system actions (L=4), or the last l system actions, where $l < L$.

3.3 Combining Sentence and Context

We study in this paper two ways of combining the sentence $\mathbf{s}_t$ and the context $\mathbf{h}_t$ representations. The first straightforward way is to apply a non linear function to their weighted sum:

$$\hat{\mathbf{h}}_{\mathbf{t}} = tanh(\mathbf{Ws} \cdot \mathbf{s}_t + \mathbf{Wc} \cdot \mathbf{h}_t) \tag{5}$$

The second way is to let the sentence representation be the last input to the LSTM network, then $\hat{\mathbf{h}}_{\mathbf{t}} = \mathbf{h}_t$.

For classification a softmax layer is used for each prediction:

$$P(Y = k|\hat{h}, W, b) = \frac{e^{(W_k \hat{h} + b_k)}}{\sum_{k\prime} e^{(W_{k\prime} \hat{h} + b_{k\prime})}} \tag{6}$$

where k is the index of the output neuron representing one class. For dialogue act classification k is one of the possible values: inform, request, offer, ... etc. For the slot prediction k is either 0 for absent or 1 for present. For slot-value prediction k will correspond to one of the possible values for each slot. For instance, for the slot price-range the possible values are cheap, moderate, expensive and dontcare. The result of the prediction is the most probable class:

$$\hat{y} = \text{argmax}_k(P(Y = k|\hat{h}, W, b)) \tag{7}$$

The back-propagation optimisation is done by minimising the negative log-likelihood loss function through stochastic gradient descent.

4 Experimental Evaluation

In this section we introduce the corpora, and describe the experiments performed and the evaluation metrics used.

4.1 Corpora

Experimental evaluation used two similar datasets: DSTC2 (Henderson et al., 2014b) and In-car (Tsiakoulis et al., 2012). Both corpora were collected using a spoken dialogue system which provides restaurant information system for the city of Cambridge. Users can specify restaurant suggestions by area, price-range and food type and can then query the system for additional restaurant specific information such as phone number, post code and address. The first dialogue corpus was released for the dialogue state tracking challenge and we use here the semantic annotations that were also provided [2]. The trainset has 2118 dialogues and 15611 turns in total while the testset has 1117 dialogues and 9890 turns in total.

The second corpus contains dialogues collected under various noisy in-car conditions. In a stationary car with the air conditioning fan on and off, in a moving car and in a car simulator (Tsiakoulis et al., 2012) [3]. The trainset has 1508 dialogues and 10532 turns in total and the testset has 641 dialogues and 4861 turns in total. Because of the noise, the average word error rate (WER = 37%) is significantly higher than for DSTC2 (around 29%).

4.2 Hyperparameters and Training

Dropout was used on the penultimate layers of both the CNN and the LSTM networks to prevent co-adaptation of hidden units by randomly dropping out a proportion of the hidden units during forward propagation (Hinton et al., 2012). The models were implemented in Theano (Bastien et al., 2012). We

[2] The DSTC2 corpus is publicly available in: `http://camdial.org/~mh521/dstc/`

[3] This corpus has been obtained in an industry funded project and therefore it is not available for public use.

used filter windows of 3, 4, and 5 with 100 feature maps each for the CNN. A dropout rate of 0.5 and a batch size of 50 was employed, 10% of the trainset was used as validation set and early stopping was adopted. Training is done through stochastic gradient descent over shuffled mini-batches with Adadelta update rule (we used an adadelta decay parameter of 0.95). To initialise the models, GloVE word vectors were used (Pennington et al., 2014) with a dimension $d = 100$. System-action word-embeddings are tuned during training, instead hypothesis word-embeddings are not because of the heavy computations.

4.3 Experiments

Step I: Joint classification of dialogue-acts and slots: We evaluated five different model configurations for the joint classification of dialogue-acts and presence or absence of slots.

- **CNN**: the softmax layers for the joint classification of dialogue acts and slots are connected directly to the CNN sentence representation with no context.
- **CNN+LSTM**: we study the influence of context by considering the previous system actions (Section 3.2, Eq. 5), here we study the different context length, by using a context window of 1, 4, and all the previous system actions, namely **CNN+LSTM_w1**, **CNN+LSTM_w4** and **CNN+LSTM_w** respectively.
- **LSTM_all**: Finally, we study the impact of long distance dependencies, by using mainly the LSTM model, with the previous system actions as input, but we inject the sentence representation as the last LSTM input.

Step II: Classification of slot value pairs: We select the best model in step I for predicting the presence of slots, then for each slot present we predict the value, by using again the best architecture from the previous step.

4.4 Evaluation Metrics

We evaluate the performance of our models by using the conventional metrics for classification, namely accuracy, precision, recall and F-measure (F1-score).

In addition, we used the ICE score (Eq. 8) between the hypotheses and the reference semantics (ie. ground-truth) to measure the overall quality of the distribution returned by the models(Thomson et al., 2008). Let U be the number of utterances and W be the number of available semantic items. Given $u = 1..U$ and $w = 1...W$, let:

$$c_{uw} = \begin{cases} p, \text{ the confidence assigned to the hypothesis that the } w^{th} \text{ semantic item is part of utterance } u, \\ 0, \text{ if none was assigned.} \end{cases}$$

$$\delta_{uw} = \begin{cases} 1, \text{ if the } w^{th} \text{ item is in the reference semantics for } u, \\ 0, \text{ otherwise} \end{cases}$$

and $N = \sum_{uw} \delta_{uw}$, be the total number of semantic items in the reference semantics.

$$\text{ICE} = \frac{1}{N_w} \sum - \log(\delta_{uw} c_{uw} + (1 - \delta_{uw})(1 - c_{uw}) \tag{8}$$

5 Results and Discussion

In this section we report the results on DSTC2 and In-car dialogue corpora.

Step I: Joint classification of dialogue-acts and slots: For this step, the classifiers must predict jointly 14 dialogue acts and 5 slots for the DSTC2 dataset as well as 14 dialogue acts and 7 slots for the In-car dataset. We evaluate both (i) using 10 fold cross-validation on the trainsets and (ii) on the corpora' testsets.

Table 1 shows the 10 fold cross-validation results on both corpora. These results suggest that for DTSC2, the context representation is not significantly impacting the prediction. Although, the model with a window of 4 ,**CNN+LSTM_w4**, improves slightly the accuracy and f1-score. On the In-car dataset, however, including the context does help to disambiguate the semantic predictions from ill-formed hypotheses. This is expected, since this data set has a much higher error rate and hence higher levels of confusion in the ASR output. Although there is no significant difference on the f1-score when using the immediate previous system act ($w1$) or a longer context, **CNN+LSTM_w** gives a better accuracy and a lower ICE score on this dataset.

Corpus	Metric	CNN	CNN+LSTM		
-	-	-	w1.	w4	w
DSTC2	acc.	$96.1\% \pm 0.002$	$95.97\% \pm 0.003$	$\mathbf{96.11}\% \pm 0.002$	$95.9\% \pm 0.003$
	P.	$90.17\% \pm 0.007$	$89.33\% \pm 0.007$	$89.77\% \pm 0.004$	$89.21\% \pm 0.008$
	R.	$85.61\% \pm 0.009$	$85.66\% \pm 0.007$	$86.40\% \pm 0.006$	$85.96\% \pm 0.006$
	F1	$87.8\% \pm 0.007$	$87.43\% \pm 0.006$	$\mathbf{88.03}\% \pm 0.004$	$87.53\%0.005$
	ICE	$\mathbf{0.245 \pm 0.013}$	0.275 ± 0.02	0.271 ± 0.02	0.277 ± 0.02
In-car	acc.	$90.45\% \pm 0.005$	$91.66\% \pm 0.003$	$91.49\% \pm 0.007$	$\mathbf{91.77}\% \pm 0.04$
	P.	$83.87\% \pm 0.01$	$84.31\% \pm 0.01$	$84.16\% \pm 0.01$	$83.89\% \pm 0.01$
	R.	$71.57\% \pm 0.007$	$74.91\% \pm 0.005$	$74.6\% \pm 0.02$	$74.76\% \pm 0.01$
	F1	$76.96\% \pm 0.008$	$\mathbf{79,16}\% \pm \mathbf{0.003}$	$78.85\% \pm 0.01$	$78.83\% \pm 0.007$
	ICE	0.498 ± 0.0013	0.457 ± 0.02	0.459 ± 0.03	$\mathbf{0.448 \pm 0.02}$

Table 1: 10 fold cross-validation evaluation of step I, the joint classification of dialogue acts and slots. Here we study the impact of the context by comparing **CNN** and **CNN+LSTM**.

Table 2 shows the results on the test sets. Consequently, when evaluating on the DSTC2 test set, a window of 4 ($w4$), performs slightly better than other window sizes and better than the simple **CNN** model. On the In-car testset, a context window of 4 outperforms all the other settings: **CNN+LSTM**. However, on this test set using the sentence representation as the last input to the LSTM context neural network (section 3.3) improves the f1-score and reduces the ICE error.

Corpus	Metric	CNN	CNN+LSTM			LSTM_all
-	-	-	w1.	w4	w	-
DSTC2	acc.	96.03%	95.79%	95.79%	95.69%	95.59%
	P.	89.73%	88.69%	88.95%	88.38%	88.15%
	R.	84.74%	85.09%	86.02%	85.96%	84.76%
	F1	87.14%	86.83%	$\mathbf{87.43}\%$	87.12%	86.42%
	ICE	$\mathbf{0.268}$	0.278	0.292	0.297	0.308
In-car	acc.	$\mathbf{87.60}\%$	82.19%	82.25%	82.14%	82.3%
	P.	69.96%	79.52%	79.29%	80.25%	78.12%
	R.	62.14%	71.09%	71.59%	70.9%	74.04%
	F1	65.53%	74.89%	75.15%	75.02%	$\mathbf{75.9}\%$
	ICE	1.332	1.344	1.333	1.421	$\mathbf{1.106}$

Table 2: Evaluation of the Step I on DSTC2 and In-car testsets. We also compare two ways of combining sentence and context representation: **CNN+LSTM** models (combining sentence and context representation through a non linear function) and **LSTM_all** model (embedding the sentence representation into the context model).

Step II: Prediction of slot value pairs For evaluating Step II, we selected the best model obtained during the 10-fold cross-validation experiments in terms of F1 score. For both corpora, this was the **CNN+LSTM_w4** configuration. For DSTC2, it was the 4^{th}-fold crossvalidation with $Acc = 90.42\%$,

$F1 = 88.69\%$ and ICE $= 0.251$. For In-car, it was the 5^{th}-fold crossvalidation with $Acc = 93.13\%$, $F1 = 81.49\%$ and ICE $= 0.393$. We used these models to classify whether a given slot appears in a given hypothesis or not. Then for that slot, we train another **CNN+LSTM_w4** classifier for predicting its values. In the In-car corpus the slot ”type” has only one possible value ”restaurant”. Similarly, the slot ”task” can only be the value ”find”. For these slots with only one value, we report values using the model of Step I, since it is enough to detect the slot in the utterance.

	DSTC2					In-car				
Slot	Acc.	P.	R.	F1	ICE	Acc.	P.	R.	F1	ICE
Slot[4]	95.29%	90.89%	95.72%	93.24%	0.478	89.92%	74.73%	61.56%	67.51%	0.743
Area	91.77%	92.66%	92.83%	92.74%	0.563	72.03%	72.56%	74.28%	73.41%	1.676
Food	71.37%	73.19%	76.02%	74.58%	1.989	66.46%	64.27%	68.70%	66.41%	2.309
Price	94.62%	91.33%	94.49%	92.89%	0.729	93.96%	88.77%	92.03%	90.37%	0.632
This[5]	98.70%	96.79%	93.92%	95.33%	0.113	97.16%	96.14%	84.72%	90.07%	0.214
Type	-	-	-	-	-	95.56%	95.09%	86.69%	90.69%	0.290
Task	-	-	-	-	-	97.12%	83.24%	64.93%	72.95%	0.175
Mean	90.35%	88.97%	90.60%	89.76%	0.774	87.47%	82.11%	76.13%	78.77%	0.863
St.Dev.	0.109	0.091	0.082	0.085	0.715	0.128	0.121	0.118	0.112	0.821

Table 3: Evaluation of the step II: the slot-value pairs classification on DSTC2 and In-car.

Given that there is no domain specific delexicalisation, the models achieve a good level of performance overall (Table 3). Note that the slot ”food” has 74 possible values in DSTC2 and 25 in In-car. Hence, this slot has much higher cardinality than all the other slots.

Overall performance A baseline for assessing overall performance is provided by the model presented in (Henderson et al., 2012), in which the vector representation is obtained by summing up the frequency of n-grams extracted from the 10-best hypotheses, weighted by their confidence scores. Here we compare our performance against Henderson’s model with and without context features, namely WNGRAMS+Ctxt and WNGRAMS repectively. Henderson reported his results on the In-car dataset. A similar model, namely SLU1, was evaluated on DSTC2 in (Williams, 2014). Both implementations consist of many binary classifiers for dialogue act and slot-value pairs.

Corpus	Model	F1	ICE
DSTC2	SLU1 (Williams, 2014)	80.2%	1.943
	CNN+LSTM_w4	**83.59%**	**0.758**
In-car	WNGRAMS (Henderson et al., 2012)	70.8%	1.76
	WNGRAMS+Ctxt (Henderson et al., 2012)	**74.2%**	1.497
	CNN+LSTM_w4	73.06%	**1.106**

Table 4: Overall performance of the setting CNN+LST_w4 semantic decoder.

In terms of the ICE score, the model CNN+LSTM W4 outperforms all the baselines (Table 4). In terms of the F1 score, the model significantly outperforms the SLU1 and WNGRAMS baselines. However it is slightly worse than WNGRAMS+Ctxt, which has been enhanced with context features on In-car. Remember however, that our model uses only word-embeddings for automatically generating sentence and context representations without having any manually designed features or using explicit application specific semantic dictionaries.

[4]”Slot” is used when no value is given for the slot (e.g., ”What kind of food do they serve?”/request(slot=food)).

[5]”This” is used for annotating elliptical utterances (e.g., ”I dont care”/inform(this=’dontcare’)).

6 Conclusion and Future Work

This paper has presented a deep learning architecture for semantic decoding in spoken dialogue systems that exploits semantically rich distributed word vectors. We compared different models for combining sentence and context representations. We found that context representations significantly impact slot F-measure on ASR hypotheses generated under very noisy conditions. The combination of sentence and context representations, with a context window of 4 words, outperforms all the baselines in terms of the ICE score. In terms of the F1 scores, our model outperforms the baseline on the DSTC2 corpus and the baseline without manually designed features on the In-car corpus. Although the F-score of our model does not outperforms the baseline enriched with context features on the In-car corpus, the proposed model remains competitive, especially considering that our model requires no manually designed features or application specific semantic dictionaries.

7 Future Work

Semantic distributed vector representations can be used for detecting similarity between domains. As future work, we want to study the adoption of the sentence and the contex representations generated in the Step I (i.e., the joint prediction of dialogue act and slots) within a *Topic Management* in multi-domain dialogue systems. The Topic Manager is in charge of detecting the domain and the intention behind users' utterances. Furthermore, it would be interesting to study these embeddings for domain adaptation on potentially open-domains.

Acknowledgments

This research was partly funded by the EP-SRC grant EP/M018946/1 Open Domain Statistical Spoken Dialogue Systems. The data used in this paper was produced in an industry funded project and it is not available for public use.

References

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. 2012. Theano: new features and speed improvements. nips workshop on deep learning and unsupervised feature learning.

Asli Celikyilmaz and Dilek Hakkani-Tur. 2010. Convolutional neural network based semantic tagging with entity embeddings. *genre*.

Yun-Nung Chen and Xiaodong He. 2015. Learning bidirectional intent embeddings by convolutional deep structured semantic models for spoken language understanding. In *Extended Abstract of The 29th Annual Conference on Neural Information Processing Systems–Machine Learning for Spoken Language Understanding and Interactions Workshop (NIPS-SLU)*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Anoop Deoras and Ruhi Sarikaya. 2013. Deep belief network based semantic taggers for spoken language understanding. In *INTERSPEECH*, pages 2713–2717.

Emmanuel Ferreira, Bassam Jabaian, and Fabrice Lefèvre. 2015. Online adaptative zero-shot learning spoken language understanding using word-embedding. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5321–5325. IEEE.

Matthew Henderson, Milica Gašić, Blaise Thomson, Pirros Tsiakoulis, Kai Yu, and Steve Young. 2012. Discriminative Spoken Language Understanding Using Word Confusion Networks. In *Spoken Language Technology Workshop, 2012. IEEE.*

M. Henderson, B. Thomson, and S. J. Young. 2014a. Word-based Dialog State Tracking with Recurrent Neural Networks. In *Proceedings of SIGdial*.

Matthew Henderson, Blaise Thomson, and Jason Williams. 2014b. The second dialog state tracking challenge. In *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, volume 263.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2015. Multi-domain Dialog State Tracking using Recurrent Neural Networks. In *Proceedings of ACL*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

R. Sarikaya, G. E. Hinton, and B. Ramabhadran. 2011. Deep belief nets for natural language call-routing. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5680–5683, May.

Ruhi Sarikaya, Geoffrey E Hinton, and Anoop Deoras. 2014. Application of deep belief networks for natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):778–784.

Blaise Thomson, Kai Yu, Milica Gasic, Simon Keizer, Francois Mairesse, Jost Schatzmann, and Steve J Young. 2008. Evaluating semantic-level confidence scores with multiple hypotheses. In *INTERSPEECH*, pages 1153–1156.

Pirros Tsiakoulis, Milica Gašic, Matthew Henderson, Joaquin Planells-Lerma, Jorge Prombonas, Blaise Thomson, Kai Yu, Steve Young, and Eli Tzirkel. 2012. Statistical methods for building robust spoken dialogue systems in an automobile. *Proceedings of the 4th applied human factors and ergonomics*.

Gökhan Tür, Anoop Deoras, and Dilek Hakkani-Tür. 2013. Semantic parsing using word confusion networks with conditional random fields. In *INTERSPEECH*, pages 2579–2583.

Jason D Williams. 2014. Web-style ranking and slu combination for dialog state tracking. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 282–291.

Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. 2013. Recurrent neural networks for language understanding. In *INTERSPEECH*, pages 2524–2528.

Deyu Zhou and Yulan He. 2011. Learning conditional random fields from unaligned data for natural language understanding. In *European Conference on Information Retrieval*, pages 283–288. Springer.

Su Zhu, Lu Chen, Kai Sun, Da Zheng, and Kai Yu. 2014. Semantic parser enhancement for dialogue domain extension with little data. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 336–341. IEEE.