# A Hierarchical Neural Model for Learning Sequences of Dialogue Acts

**Quan Hung Tran** and **Ingrid Zukerman** and **Gholamreza Haffari**
Faculty of Information Technology, Monash University
Clayton, VICTORIA 3800, Australia
{hung.tran,ingrid.zukerman,gholamreza.haffari}@monash.edu

## Abstract

We propose a novel hierarchical Recurrent Neural Network (RNN) for learning sequences of Dialogue Acts (DAs). The input in this task is a sequence of utterances (i.e., conversational contributions) comprising a sequence of tokens, and the output is a sequence of DA labels (one label per utterance). Our model leverages the hierarchical nature of dialogue data by using two nested RNNs that capture long-range dependencies at the dialogue level and the utterance level. This model is combined with an attention mechanism that focuses on salient tokens in utterances. Our experimental results show that our model outperforms strong baselines on two popular datasets, Switchboard and MapTask; and our detailed empirical analysis highlights the impact of each aspect of our model.

## 1 Introduction

The sequence-labeling task involves learning a model that maps an input sequence to an output sequence. Many NLP problems can be treated as sequence-labeling tasks, e.g., part-of-speech (PoS) tagging (Toutanova et al., 2003; Toutanova and Manning, 2000), machine translation (Brown et al., 1993) and automatic speech recognition (Gales and Young, 2008). Recurrent Neural Nets (RNNs) have been the workhorse model for many NLP sequence-labeling tasks, e.g., machine translation (Sutskever et al., 2014) and speech recognition (Amodei et al., 2015), due to their ability to capture long-range dependencies inherent in natural language.

In this paper, we propose a hierarchical RNN for labeling a sequence of utterances (i.e., contributions) in a dialogue with their Dialogue Acts (DAs). This task is particularly useful for dialogue systems, as knowing the DA of an utterance supports its interpretation, and the generation of an appropriate response. The DA classification problem differs from the aforementioned tasks in the structure of the input and the immediacy of the output. The input in these tasks is a sequence of tokens, e.g., a sequence of words in PoS tagging; while in DA classification, the input is hierarchical, i.e., a conversation comprises a sequence of utterances, each of which has a sequence of tokens (Figure 1). In addition, to be useful for dialogue systems, the DA of an utterance must be determined immediately, hence a bi-directional approach is not feasible.

As mentioned above, RNNs are able to capture long-range dependencies. This ability was harnessed by Shen and Lee (2016) for DA classification. However, they ignored the conversational dimension of the data, treating the utterances in a dialogue as separate instances — an assumption that results in loss of information. To overcome this limitation, we designed a two-layer RNN model that leverages the hierarchical nature of dialogue data: an outer-layer RNN encodes the conversational dimension, and an inner-layer RNN encodes the utterance dimension.

One of the difficulties of sequence labeling is that different elements of an input sequence have different degrees of importance for the task at hand (Shen and Lee, 2016), and the noise introduced by less important elements might degrade the performance of a labeling model. To address this problem, we incorporate into our model the attention mechanism described in (Shen and Lee, 2016), which has yielded performance improvements in DA classification compared to traditional RNNs.

Our empirical results show that our *hierarchical RNN model with an attentional mechanism* out-

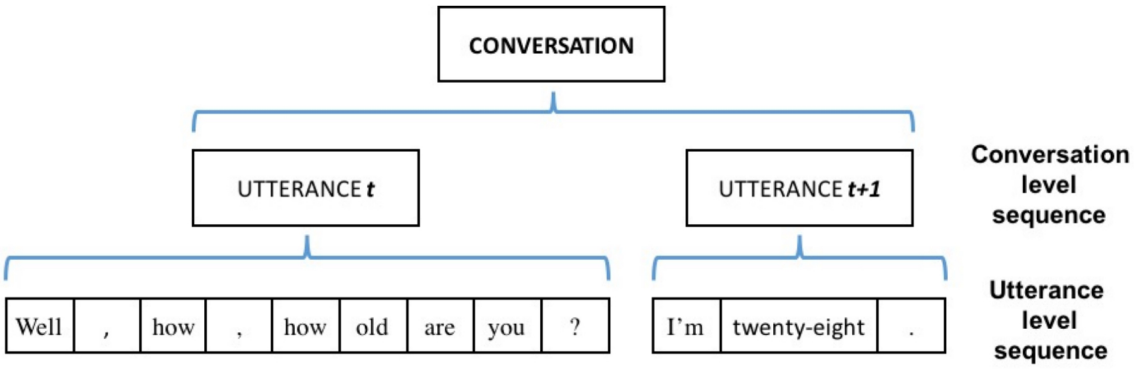


Figure 1: Switchboard data example.

performs strong baselines on two popular datasets: Switchboard (Jurafsky et al., 1997; Stolcke et al., 2000) and MapTask (Anderson et al., 1991). In addition, we provide an empirical analysis of the impact of the main aspects of our model on performance: utterance RNN, conversation RNN, and information source for the attention mechanism.

This paper is organised as follows. In the next section, we discuss related research in DA classification. In Section 3, we describe our RNN. Our experiments and results are presented in Section 4, followed by our analysis and concluding remarks.

## 2 Related Research

**Independent DA classification.** In this approach, each utterance is treated as a separate instance, which allows the application of general classification algorithms. Julia *et al.* (2010) employed a Support Vector Machine (SVM) with n-gram features obtained from an utterance-level Hidden Markov Model (HMM) to ascribe DAs to audio signals and textual transcriptions of the MapTask corpus. Webb *et al.* (2005) used a similar approach, employing cue phrases as features.

**Sequence-based DA classification.** This approach takes advantage of the sequential nature of conversations. In one of the earliest works in DA classification, Stolcke *et al.* (2000) used an HMM with a trigram language model to classify DAs in the Switchboard corpus, achieving an accuracy of 71.0%. In this work, the trigram language model was employed to calculate the symbol emission probability of the HMM. Surendran *et al.* (2006) also used an HMM, but employed output symbol probabilities produced by an SVM classifier, instead of emission probabilities obtained from a trigram language model. More recently, the Recurrent Convolutional Neural Network model proposed by Kalchbrenner and Blunsom (2013) achieved an accuracy of 73.9% on the Switchboard corpus. In this work, a Convolutional Neural Network encodes each utterance into a vector, which is then treated as input to a conversation-level RNN. The DA is then classified using a softmax layer applied on top of the hidden states of the RNN.

**Attention in Neural Models.** Attentional Neural Models have been successfully applied to sequence-to-sequence mapping tasks, notably machine translation and DA classification. Bahdanau *et al.* (2014) proposed an attentional encoder-decoder architecture for machine translation. The encoder encodes the input sequence into a sequence of hidden vectors; the decoder decodes the information stored in the hidden sequence to generate the output; and the attentional mechanism is used to summarize a sentence into a context vector dynamically, helping the decoder decide which part of the sequence to attend to when generating a target word. As mentioned above, Shen and Lee (2016) employed an attentional RNN for independent DA classification; they achieved an accuracy of 72.6% on textual transcriptions of the Switchboard corpus.

## 3 Model Description

Suppose we have a sequence of observations $\boldsymbol{o} := \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_m\}$ and the corresponding sequence of labels $\boldsymbol{y} := \{y_1, y_2, \ldots, y_m\}$, where each observation $\boldsymbol{o}_t$ is a sequence. Our hierarchical-attentional model, denoted *HA-RNN*, learns the conditional probability $P(\boldsymbol{y}|\boldsymbol{o})$ relating the ob-

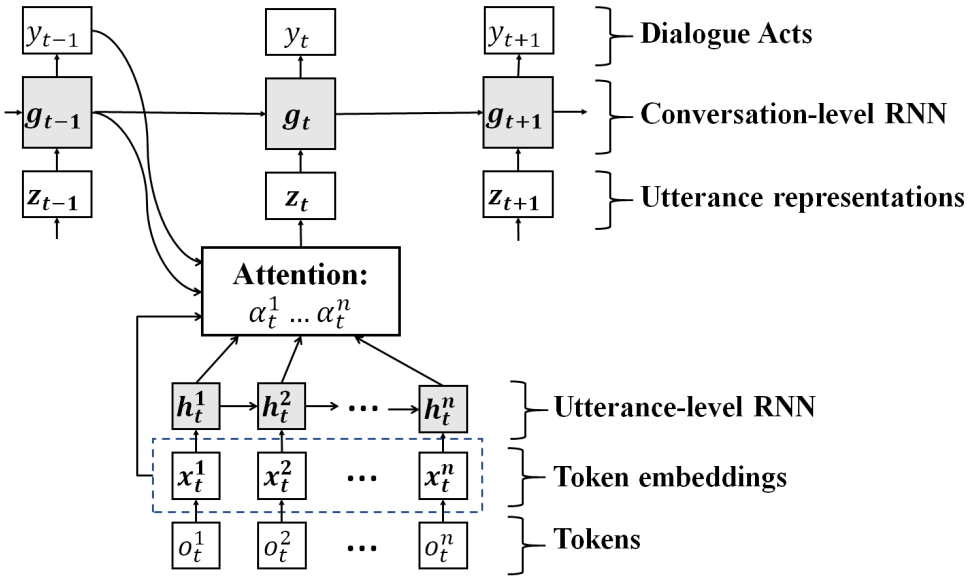


Figure 2: *HA-RNN* – Hierarchical-attentional RNN model.

served sequence to its label sequence, based on the following decomposition:

$$P(\{y_1, y_2, ..., y_m\}|\{\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_m\}) = \prod_{t=1}^{m} P(y_t|\boldsymbol{y}_{<t}, \boldsymbol{o}_{\leq t}) \quad (1)$$

Note that our model conditions on the full history, rather than a finite history as done in Markov models, such as maximum entropy Markov models (McCallum et al., 2000).

We employ neural networks to model the constituent conditional distributions. Our model comprises three main elements (Figure 2): (1) an *utterance-level RNN* that encodes the information within the utterances; (2) an *attentional mechanism* that highlights the important parts of an input utterance, and summarizes the information within the utterance into a real-valued vector; and (3) a *conversation-level RNN* that encodes the information of the whole dialogue sequence. As discussed in Section 1, our hierarchical-RNN design was motivated by the structure of the input data, while the attentional mechanism has proven to be effective in DA classification (Shen and Lee, 2016).

**Utterance-level RNN.** This RNN was implemented using LSTM (Hochreiter and Schmidhuber, 1997; Graves, 2013). First, an embedding matrix maps each token (e.g., word or punctuation marker) into a dense vector representation. Let us denote the sequence of tokens in the $t$-th utterance as $\boldsymbol{o}_t := \{o_t^1, o_t^2, \ldots, o_t^n\}$, which is mapped into the sequence of embedding vectors $\boldsymbol{x}_t := \{\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, \ldots, \boldsymbol{x}_t^n\}$ using the token embedding table $\boldsymbol{w}$:

$$\boldsymbol{x}_t^i = \boldsymbol{e}_{\boldsymbol{w}}(o_t^i) \quad (2)$$

The utterance RNN then takes as input this sequence of vectors, and produces a sequence of corresponding hidden vectors $\boldsymbol{h}_t = \{\boldsymbol{h}_t^1, \boldsymbol{h}_t^2, \ldots, \boldsymbol{h}_t^n\}$, which capture the information within the tokens, and put the tokens in their sentential context:

$$\boldsymbol{h}_t^i = \text{RNN}_{utter}(\boldsymbol{h}_t^{i-1}, \boldsymbol{x}_t^i) \quad (3)$$

The parameters of the utterance RNN and the token embeddings are learned during training.

**Attentional mechanism.** This mechanism summarizes the hidden vectors of the utterance-level RNN into a single vector representing the whole utterance. The attention vector is a sequence of positive numbers that sum to 1, where each number corresponds to a token in an utterance, and represents the importance of the token for understanding the DA associated with the utterance. The final representation $\boldsymbol{z}_t$ of the $t$-th utterance is the sum of the corresponding elements of its hidden vectors weighted by attention weights:

$$\boldsymbol{z}_t = \sum_i \alpha_t^i \boldsymbol{h}_t^i \quad (4)$$

We posit that the main factors for determining the importance of a token for DA classification are: (1) the meaning of the token, as represented

by its embedding vector; and (2) the full context of the conversation, particularly the previous DA. For example, if the DA of an utterance is *Yes-No-Question*, and there is a "yes" or "no" token in the next utterance, this token is likely to be important. Equation 5 integrates these factors to compute attention scores:

$$s_t^i = \boldsymbol{U} \cdot \tanh\big(W^{(\text{in})} \cdot \boldsymbol{x}_t^i + W^{(\text{co})} \cdot \boldsymbol{g}_{t-1} + \boldsymbol{e_a}(y_{t-1}) + \boldsymbol{b}^{(\text{in})}\big) \quad (5)$$

where vector $\boldsymbol{e_a}(y_{t-1})$ denotes the embedding of the previous DA, which is similar to the embedding of tokens; and vector $\boldsymbol{g}_{t-1}$ is the previous hidden vector of the conversation-level RNN, detailed below, which summarizes the conversation so far. $W^{(\text{in})}$ and $W^{(\text{co})}$ are parameter matrices for the input tokens and the conversational context respectively, and $\boldsymbol{U}$ and $\boldsymbol{b}^{(\text{in})}$ are parameter vectors — all of which are learned during training. The scores $s_t^i$ are mapped into a probability vector by means of a softmax function:

$$\boldsymbol{\alpha}_t = \text{softmax}(\boldsymbol{s}_t) \quad (6)$$

**Conversation-level RNN.** This RNN is structurally similar to the utterance-level RNN. The input to the conversation-level RNN is the sequence of vectors $\boldsymbol{z}$ generated for the utterances in a conversation, which is then encoded by the RNN into a sequence of hidden vectors $\boldsymbol{g}$:

$$\boldsymbol{g}_t = \text{RNN}_{convers}(\boldsymbol{g}_{t-1}, \boldsymbol{z}_t) \quad (7)$$

This information is then used in the generation of the output DA:

$$y_t|\boldsymbol{y}_{<t}, \boldsymbol{o}_{\leq t} \sim \text{softmax}(\boldsymbol{W}^{(\text{out})} \cdot \boldsymbol{g}_t + \boldsymbol{b}^{(\text{out})}) \quad (8)$$

where the matrix $\boldsymbol{W}^{(\text{out})}$, vector $\boldsymbol{b}^{(\text{out})}$ and the parameters of the conversation-level network $\text{RNN}_{convers}$ are learned during the training.

During testing, ideally a given sequence of observed utterances $\boldsymbol{o}$ should be decoded to a label sequence $\boldsymbol{y}$ that maximizes the conditional probability $P(\boldsymbol{y}|\boldsymbol{o})$ according to the model. However, finding the highest-scoring label sequence is a computationally hard problem, since the conversation-level RNN does not lend itself to dynamic programming. Therefore, we employ a greedy decoding approach, where, going left-to-right, at each step we choose the $y_t$ with the highest probability in the local DA distribution. This method is common practice in sequence-labeling RNNs, e.g., in neural machine translation (Bahdanau et al., 2014; Sutskever et al., 2014; Luong et al., 2015).

## 4 Experiments

### 4.1 Data sets

We tested our models on the Switchboard corpus (Jurafsky et al., 1997; Stolcke et al., 2000) and the MapTask corpus (Anderson et al., 1991) — two popular datasets used for DA classification. At this stage of our research, we consider only transcriptions of the conversations in both corpora (the incorporation of phonetic input (Taylor et al., 1998; Wright Hastie et al., 2002; Julia et al., 2010) is the subject of future work). Thus, we compare our results only with those obtained by systems that employ transcriptions exclusively.

**Switchboard corpus.** This corpus contains DA-annotated transcriptions of 1155 telephone conversations with no specific topic, which have an average of 176 utterances. Originally, there were approximately 226 DA tags in the corpus, but in the DA classification literature, the tags are usually clustered into 42 tags.[1] Table 1(a) shows percentages of the seven most frequent tags in the data. Following (Stolcke et al., 2000), in our experiments we use 1115 conversations for training, 21 for development and 19 for testing.

**MapTask corpus.** This is a richly annotated corpus that comprises 128 dialogues about instruction following, containing 212 utterances on average. Each conversation has an instruction giver and an instruction follower. The instruction giver gives directions with reference to a map, which the instruction follower must follow. The MapTask corpus has 13 DA tags, including the "unclassifiable" tag. Table 1(b) shows percentages of the seven most frequent tags in the data. We randomly split this data into 80% training, 10% development and 10% test sets, which contain 103, 12 and 13 conversations respectively.

### 4.2 Results

We experimented with different embedding sizes and hidden layer dimensions for our model *HA-RNN*, and selected the following, which yielded

[1] The official manual stated that there were originally 220 tags. We follow the tag-clustering procedure by Christopher Potts described in `compprag.christopherpotts.net/swda.html`.

| DA tag | Example | Percentage |
|---|---|---|
| *Statement-non-opinion* | I'm twenty-eight | 36% |
| *Acknowledge (Backchannel)* | Uh-huh | 19% |
| *Statement-opinion* | I think it's great | 13% |
| *Agree/Accept* | That's exactly it | 5% |
| *Abandoned or Turn-Exit* | So, | 5% |
| *Appreciation* | I can imagine. | 2% |
| *Yes-No-Question* | Do you? | 2% |

(a) Switchboard

| DA tag | Example | Percentage |
|---|---|---|
| *Acknowledge* | Mmhmm | 21% |
| *Instruct* | And we're finished | 16% |
| *Reply_y* | Yeah | 12% |
| *Explain* | I've got a bridge | 8% |
| *Check* | Is that it | 8% |
| *Ready* | And then | 8% |
| *Align* | See what i mean | 7% |

(b) MapTask

Table 1: Seven most frequent DAs and examples for (a) Switchboard and (b) MapTask.

| Model | Accuracy |
|---|---|
| *RCNN* | 73.9% |
| *RNN-Attentional-C* | 72.6% |
| *HMM-trigram-C* | 71.0% |
| *HA-RNN* | **74.5%** |

Table 2: Performance on Switchboard.

| Model | Accuracy |
|---|---|
| *HMM-trigram-C* | 52.3% |
| *Random Forest* | 52.5% |
| *Random Forest + prev DA* | 55.3% |
| *HA-RNN* | **63.3%** |

Table 3: Performance on MapTask.

the best performance with reasonable run times. The word-embedding size was set to 250, and the DA-embedding size to 180. The hidden dimension of the utterance-level RNN was set to 160, and the hidden dimension of the conversation-level RNN was set to 250. Our model was implemented with the CNN package.[2] During training, the negative log-likelihood was optimized using Adagrad (Duchi et al., 2011), with dropout rate 0.5 to prevent over-fitting (Srivastava et al., 2014). Training terminated when the log-likelihood of the development set did not improve. As mentioned in Section 3, during testing, the sequence of output labels was generated with greedy decoding. Statistical significance was computed on the MapTask test data using McNemar's test with $\alpha = 0.05$ (we could not compute statistical significance for the Switchboard results, because they were obtained from the literature, and we did not have access to per-conversation labels).

**Switchboard.** We compare our model's performance with that of the following strong baselines: (*RCNN*) the recurrent convolutional neural network model from (Kalchbrenner and Blunsom, 2013); (*RNN-Attentional-C*) the attention-based RNN classifier from (Shen and Lee, 2016); and (*HMM-trigram-C*) the HMM-based classifier from (Stolcke et al., 2000).

The results in Table 2 show that our model outperforms these baselines.[3] The higher accuracy of our model compared to classifier-based approaches (i.e., *RNN-Attentional-C* and *HMM-trigram-C*) confirms that taking into account dependencies among the DAs through the conversation-level RNN improves accuracy. Furthermore, the better performance of our model compared to *RCNN* shows that summarizing utterances with an RNN augmented with an attention architecture is more effective than using a convolution architecture for DA sequence labeling.

**MapTask.** Due to the unavailability of standard training/development/test sets for this dataset, we compare the results obtained by our model with those obtained by our implementation of the following independent DA classifiers: *HMM-trigram-C* (Stolcke et al., 2000); *Random Forest* – an instance-based random forest classifier; and *Random Forest + prev DA* – a random forest classifier that uses the previous DA tag.

The results in Table 3 show that our model outperforms these baselines (statistically significant). These results reinforce the insights from the Switchboard corpus, whereby taking into account conversational dependencies between DAs substantially improves DA-labeling performance.[4]

[2] `github.com/clab/cnn`.

[3] Two other works on Switchboard DA classification (Gambäck et al., 2011; Webb and Ferguson, 2010) used experimental setups that differ from ours, respectively obtaining accuracies of 77.85% and 80.72%. However, these results are not directly comparable to Stolcke *et al.*'s (2000) or ours, and are therefore excluded from our comparison.

[4] Two studies on MapTask DA classification were performed under experimental setups that differ from ours: Julia *et al.* (2010) employed *HMM+SVM* on text transcriptions and audio signals, obtaining an accuracy of 55.4% for transcriptions only. Surendran and Levow (2006) used *Viterbi+SVM*, posting a classification accuracy of 59.1% for transcriptions — the best result among systems that employ transcription data exclusively. Unfortunately, Julia *et al.*'s de-

## 5 Analysis

### 5.1 Architectural analysis

We investigate the influence of the main components of our model on performance by creating variants of our model through the addition or removal of connections or layers. We then compare the performance of these variants with that of the original model in terms of DA-classification accuracy and negative log-likelihood on the test, development and training partitions of our datasets. As done in Section 4, statistical significance is calculated for the test partitions of both datasets using McNemar's test with $\alpha = 0.05$.

**Does an RNN at the utterance level help?** To answer this question, we create a variant, denoted *woUttRNN*, where attentional coefficients are applied directly to the token embeddings. Thus, Equation 4 is changed to Equation 9:

$$\boldsymbol{z}_t = \sum_i \alpha_t^i \boldsymbol{x}_t^i \tag{9}$$

As seen in Tables 4 and 5, removing the utterance-level RNN (*woUttRNN*) reduces the accuracy and increases the negative log likelihood for the training, development and test partitions of both datasets. These changes are statistically significant for the test set.

**Which sources of information are critical for computing the attentional component?** In our main model, *HA-RNN*, we calculate the attentional signal using information from the previous DA, the previous hidden vector representation of the conversation-level RNN, and the embeddings of the tokens. To determine the contribution of the first two resources to the performance of the model, we create two variants of *HA-RNN*: *woDA2Attn*, which employs only the previous conversation-level RNN hidden vector; and *woHid2Attn*, which employs only the previous DA. Thus, in *woDA2Attn*, Equation 5 becomes Equation 10, and in *woHid2Attn*, Equation 5 becomes Equation 11:

$$s_t^i = \boldsymbol{U} \cdot \tanh(W^{(\text{in})} \cdot \boldsymbol{x}_t^i + W^{(\text{co})} \cdot \boldsymbol{g}_{t-1} + \boldsymbol{b}^{(\text{in})}) \tag{10}$$

$$s_t^i = \boldsymbol{U} \cdot \tanh(W^{(\text{in})} \cdot \boldsymbol{x}_t^i + \boldsymbol{e_a}(y_{t-1}) + \boldsymbol{b}^{(\text{in})}) \tag{11}$$

scription of their MapTask subset is not sufficient to replicate their experiment, and Surendran and Levow's data split is not accessible. Notwithstanding the difference in conditions, our model's accuracy is superior to theirs.

As seen in Tables 4 and 5, both of these resources provide valuable information, but the changes in performance due to the omission of these resources are smaller than those obtained with *woUttRNN*. Removing the DA connection (*woDA2Attn*) or the previous conversation-level RNN hidden vector (*woHid2Attn*) leads to statistically significant drops in accuracy and increases in negative log-likelihood on the test partitions of both datasets. The changes in performance with respect to the development and training sets vary across the datasets. As seen in Table 4, both models exhibit accuracy drops (and small increases in negative log-likelihood) on the Switchboard development set, but small accuracy increases (and negative log-likelihood drops) on the Switchboard training set — an indication of over-fitting. In contrast, as seen in Table 5, both models yield a negligible or no drop in accuracy on the MapTask development set, while both yield a drop in accuracy on the training set.

**How important is the RNN at the conversation level?** To answer this question, we create a variant of our *HA-RNN* model, denoted *woConvRNN*, where the recurrent connections between the units in the conversation-level RNN are removed. The LSTM basis function is calculated with a fixed vector $\boldsymbol{g}_0$ instead of the previous time step's vector. Thus Equation 7 becomes Equation 12:

$$\boldsymbol{g}_t = \boldsymbol{f}(\boldsymbol{g}_0, \boldsymbol{z}_t) \tag{12}$$

As seen in Tables 4 and 5, *HA-RNN* outperforms *woConvRNN* on the training/development/test partitions of both datasets. The difference between the performance of *HA-RNN* and *woConvRNN* is statistically significant for the test set.

**How effective are the DA connections?** We have seen that the DA connections improve our model's performance when they are used to calculate the attentional signal. However, intuitively, the previous DA can also directly provide information about the current DA. For example, it is often the case that a *Yes-No-Question* is followed by *Reply_y* or *Reply_n*. To reflect this observation, we create another model, denoted *wDA2DA*, that has an additional direct connection between the previous DA and the current DA. That is, Equation 8 becomes Equation 13:

$$y_t | \boldsymbol{y}_{<t}, \boldsymbol{o}_{\leq t} \sim \text{softmax}(\boldsymbol{W}^{(\text{out})} \cdot \boldsymbol{g}_t + \boldsymbol{e_o}(y_{t-1}) + \boldsymbol{b}^{(\text{out})}) \tag{13}$$

| | Accuracy | | | Neg log likelihood | | |
|---|---|---|---|---|---|---|
| | Test | Dev | Train | Test | Dev | Train |
| ***HA-RNN*** | **74.5%** | **76.2%** | 80.0% | **3770** | **2819** | 130333 |
| ***woUttRNN*** | 71.8% | 73.3% | 77.4% | 4542 | 3474 | 163350 |
| ***woDA2Attn*** | 72.7% | 74.3% | 80.5% | 3835 | 2932 | 127445 |
| ***woHid2Attn*** | 72.8% | 75.0% | **81.0%** | 4024 | 2917 | **124483** |
| ***woConvRNN*** | 71.8% | 74.1% | 76.7% | 4537 | 3648 | 165734 |
| ***wDA2DA*** | 71.0% | 72.7% | 79.2% | 4737 | 3884 | 150436 |

Table 4: Performance of variants of the *HA-RNN* model on Switchboard.

| | Accuracy | | | Neg log likelihood | | |
|---|---|---|---|---|---|---|
| | Test | Dev | Train | Test | Dev | Train |
| ***HA-RNN*** | **63.3%** | **61.9%** | **73.4%** | **3486** | 3228 | **18191** |
| ***woUttRNN*** | 56.9% | 58.0% | 62.2% | 3823 | 3445 | 25074 |
| ***woDA2Attn*** | 61.4% | 61.7% | 70.1% | 3539 | **3212** | 19780 |
| ***woHid2Attn*** | 62.2% | **61.9%** | 71.8% | 3487 | 3248 | 19132 |
| ***woConvRNN*** | 58.9% | 60.0% | 66.9% | 3579 | 3248 | 20961 |
| ***wDA2DA*** | 58.2% | 58.4% | 69.3% | 4014 | 3663 | 21135 |

Table 5: Performance of variants of the *HA-RNN* model on MapTask.

As seen in Tables 4 and 5, *wDA2DA* performs much worse than *HA-RNN*. We posit that this happens due to the *exposure bias* problem (Ranzato et al., 2015). That is, during training, the model has access to the correct DA of the previous utterance. However, during testing, the decoding process has access only to predicted DAs, which may lead to the propagation of errors. To quantify the effect of this problem on our model, we designed another experiment where the variants of our model can access the correct DA even during testing; the results for the test partitions of both datasets appear in Table 6.

The results in Table 6 show that exposure bias has different effects on the different variants of our model. As expected, *woDA2Attn*, which does not consider the previous DA, exhibits no change in performance between the oracle and greedy conditions. The models that employ a DA connection to compute the attention signal (*HA-RNN*, *woUttRNN*, *woHid2Attn*, *woConvRNN*) show a slight improvement in accuracy when using the correct DA as input, instead of the predicted DA. In contrast, *wDA2DA* shows large improvements when using the correct DA (3.5% on Switchboard and 6.8% on MapTask), becoming the best-performing model for both datasets. This improvement may be attributed to the direct connection between the DAs in this model, which increases the influence of previous DAs on the prediction of the current DA — previous DA predictions that are largely correct will substantially improve the performance of *wDA2DA*, while noisy DA predictions will have the opposite effect.

### 5.2 Attentional Analysis

We analyze how our model *HA-RNN* distributes attention over the tokens in an utterance in order to identify tokens in focus.

Figure 3 shows how the attentional vector highlights the most important tokens in sample utterances in the context of the DA-classification task. For example, in "yes I do", the most important token that identifies the *Reply_y* class is the token "yes", which receives most of the probability mass from the attention mechanism.

Table 7 shows the most attended tokens for four classes of DA in MapTask. We compiled these lists by computing the average attention that a token received for all the utterances in a DA class (we excluded tokens that appear less than 5 times). As shown in Table 7, both important tokens "move" and "yes" in Figure 3 appear in their respective DA columns. Two of the most common

| | Switchboard | | MapTask | |
|---|---|---|---|---|
| | **Oracle** | **Greedy** | **Oracle** | **Greedy** |
| ***HA-RNN*** | 74.6% | **74.5%** | 64.1% | **63.3%** |
| *woUttRNN* | 73.2% | 71.8% | 56.9% | 57.1% |
| *woDA2Attn* | 73.7% | 73.7% | 61.4% | 61.4% |
| *woHid2Attn* | 73.8% | 72.8% | 62.4% | 62.2% |
| *woConvRNN* | 72.2% | 71.8% | 58.9% | 58.9% |
| *wDA2DA* | **75.0%** | 71.5% | **65.0%** | 58.2% |

Table 6: Performance of oracle and greedy decoding on Switchboard and MapTask test data.


Figure 3: Sample DAs with highlighted attention vectors for MapTask.

| ***Acknowledge*** | ***Instruct*** | ***Reply_y*** | ***Reply_n*** |
|---|---|---|---|
| mmhmm | move | mmhmm | nope |
| uh-huh | continue | uh-huh | i've |
| yes | drop | yes | no |
| yeah | starting | yep | it's |
| see | pass | aye | you |
| go | reach | i've | go |
| aye | stop | yeah | don't |
| no | coming | i'm | not |
| you | go | you | haven't |
| i'm | whatever | go | just |

Table 7: Sample DA-specific high-focus tokens for MapTask.

labels, *Acknowledge* and *Reply_y*, have very similar attended tokens. In fact, many utterances in *Acknowledge* and *Reply_y* have the same text form. Thus, the distinction between the two classes is highly dependent upon the conversational context. Also, note that although *Reply_n* is not one of the most common DAs in MapTask, our model can still learn the most important tokens for this DA.

## 6 Conclusions

In this paper, we proposed a novel hierarchical RNN for learning sequences of DAs. Our model leverages the hierarchical nature of dialogue data by using two nested RNNs that capture long-range dependencies at the conversation level and the utterance level. We further combine the model with an attention mechanism to focus on salient tokens in utterances. Our experimental results show that our model outperforms strong baselines on two popular datasets: Switchboard and MapTask. In the future, we plan to address the exposure bias problem, and incorporate acoustic features and speaker information into our model.

## Acknowledgments

This research was supported in part by grant DP120100103 from the Australian Research Council.

## References

Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, et al. 2015. Deep speech 2: End-to-end speech recognition in English and Mandarin. *arXiv preprint arXiv:1512.02595*.

Anne H. Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Steven Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, et al. 1991. The HCRC MapTask corpus. *Language and speech*, 34(4):351–366.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Mark Gales and Steve Young. 2008. The application of hidden Markov models in speech recognition. *Foundations and trends in signal processing*, 1(3):195–304.

Björn Gambäck, Fredrik Olsson, and Oscar Täckström. 2011. Active learning for dialogue act classification. In *Proceedings of Interspeech 2011*, pages 1329–1332, Florence, Italy.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

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

Fatema N. Julia, Khan M. Iftekharuddin, and Atiq U. Islam. 2010. Dialog act classification using acoustic and discourse information of MapTask data. *International Journal of Computational Intelligence and Applications*, 9(4):289–311.

Daniel Jurafsky, Elizabeth Shriberg, and Debra Biasca. 1997. Switchboard SWBD-DAMSL Shallow-Discourse-Function Annotation Coders Manual, Draft 13. Technical report, Stanford University.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. *arXiv preprint arXiv:1306.3584*.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP'2015 – Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal.

Andrew McCallum, Dayne Freitag, and Fernando C.N. Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. In *ICML'00 – Proceedings of the 17th International Conference on Machine Learning*, pages 591–598, Stanford, California.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Sheng-syun Shen and Hung-yi Lee. 2016. Neural attention models for sequence classification: Analysis and application to key term extraction and dialogue act detection. *arXiv preprint arXiv:1604.00077*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Andreas Stolcke, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Dinoj Surendran and Gina-Anne Levow. 2006. Dialog act tagging with Support Vector Machines and hidden Markov models. In *Proceedings of Interspeech 2006*, pages 1950–1953, Pittsburgh, Pennsylvania.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Paul A. Taylor, Simon King, Steve D. Isard, and Helen Wright Hastie. 1998. Intonation and dialogue context as constraints for speech recognition. *Language and Speech*, 41(3-4):493–512.

Kristina Toutanova and Christopher D Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 63–70, Hong Kong.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network.

In *NAACL'2003 – Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 173–180, Edmonton, Canada.

Nick Webb and Michael Ferguson. 2010. Automatic extraction of cue phrases for cross-corpus dialogue act classification. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1310–1317, Uppsala, Sweden.

Nick Webb, Mark Hepple, and Yorick Wilks. 2005. Dialogue act classification based on intra-utterance features. In *Proceedings of the AAAI Workshop on Spoken Language Understanding*, Pittsburgh, Pennsylvania.

Helen Wright Hastie, Massimo Poesio, and Steve D. Isard. 2002. Automatically predicting dialogue structure using prosodic features. *Speech Communication*, 36(1):63–79.