A Generative Attentional Neural Network Model for Dialogue Act Classification

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

Abstract

We propose a novel generative neural network architecture for Dialogue Act classification. Building upon the Recurrent Neural Network framework, our model incorporates a new attentional technique and a label-to-label connection for sequence learning, akin to Hidden Markov Models. Our experiments show that both of these innovations enable our model to outperform strong baselines for dialogue-act classification on the MapTask and Switchboard corpora. In addition, we analyse empirically the effectiveness of each of these innovations.

1 Introduction

Dialogue Act (DA) classification is a sequence-to-sequence learning task where a sequence of utterances is mapped into a sequence of DAs. Some works in DA classification treat each utterance as an independent instance (Julia et al., 2010; Gambäck et al., 2011), which leads to ignoring important long-range dependencies in the dialogue history. Other works have captured inter-utterance relationships using models such as Hidden Markov Models (HMMs) (Stolcke et al., 2000; Surendran and Levow, 2006) or Recurrent Neural Networks (RNNs) (Kalchbrenner and Blunsom, 2013; Ji et al., 2016), where RNNs have been particularly successful.

In this paper, we present a generative model of utterances and dialogue acts which conditions on the relevant part of the dialogue history. To this effect, we use the *attention* mechanism (Bahdanau et al., 2014) developed originally for sequence-to-sequence models, which has proven effective in Machine Translation (Bahdanau et al., 2014; Luong et al., 2015) and DA classification (Shen and Lee, 2016). The intuition is that different parts of an input sequence have different levels of importance with respect to the objective, and this mechanism enables the selection of the important parts. However, the traditional attention mechanism suffers from the *attention-bias* problem (Wang et al., 2016), where the attention mechanism tends to favor the inputs at the end of a sequence. To address this problem, we propose a *gated attention* mechanism, where the attention signal is represented as a gate over the input vector.

In addition, when generating a dialogue act, we capture its direct dependence on the previous dialogue act — a reasonable source of information, which, surprisingly, has not been explored in the RNN literature for DA classification.

Our experiments show that our model significantly outperforms variants that do not have our innovations, i.e., the gated attention mechanism and direct label-to-label dependency.

2 Model Description

Assume that we have a training dataset $\mathcal{D}$ comprising a collection of dialogues, where each dialogue consists of a sequence of utterances $\{\boldsymbol{y}_t\}_{t=1}^{T}$ and the corresponding sequence of dialogue acts $\{z_t\}_{t=1}^{T}$. Each utterance $\boldsymbol{y}_t$ is a sequence of tokens, and its n-th token is denoted $y_{t,n}$.

We propose a generative neural model for dialogue $P_{\boldsymbol{\Theta}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$, which specifies a joint probability distribution over a sequence of utterances $\boldsymbol{y}_{1:T}$ and the corresponding sequence of dialogue acts $\boldsymbol{z}_{1:T}$. This generative model is then trained discriminatively by maximising the conditional log-likelihood $\sum_{(\boldsymbol{z}_{1:T},\boldsymbol{y}_{1:T})\in\mathcal{D}} \log P_{\boldsymbol{\Theta}}(\boldsymbol{z}_{1:T}|\boldsymbol{y}_{1:T})$:

$$\arg\max_{\boldsymbol{\Theta}} \sum_{(\boldsymbol{y}_{1:T},\boldsymbol{z}_{1:T})\in\mathcal{D}} \log \frac{P_{\boldsymbol{\Theta}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})}{\sum_{\boldsymbol{z}'_{1:T}} P_{\boldsymbol{\Theta}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}'_{1:T})}$$

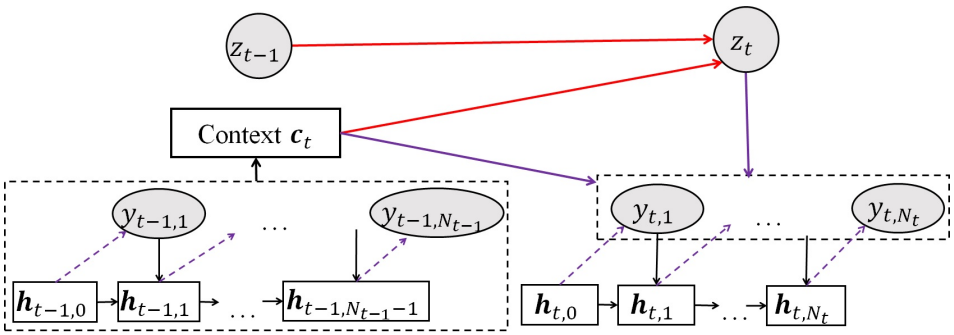

Figure 1: Graphical model representation of our model. Red connections depict dialogue-act generation (1); purple connections (dashed and continuous) depict utterance generation (2).

where $\boldsymbol{\Theta}$ represents all neural network parameters. Discriminative training is employed in order to match the use of the model for predicting dialogue acts during test time, using $\arg\max_{\boldsymbol{z}'_{1:T}} P_{\boldsymbol{\Theta}}(\boldsymbol{z}'_{1:T}|\boldsymbol{y}_{1:T})$.

The generative story of our model is as follows: (1) generate the dialogue act of the current dialogue turn conditioned on the previous dialogue act and the previous utterance $P_{\boldsymbol{\Theta}}(z_t|z_{t-1}, \boldsymbol{y}_{t-1})$; and (2) generate the current utterance conditioned on the previous utterance and the current dialogue act $P_{\boldsymbol{\Theta}}(\boldsymbol{y}_t|z_t, \boldsymbol{y}_{t-1})$. In other words, $P_{\boldsymbol{\Theta}}(\boldsymbol{z}_{1:T}, \boldsymbol{y}_{1:T})$ is decomposed as:

$$\prod_{t=1}^{T} P_{\boldsymbol{\Theta}}(z_t|z_{t-1}, \boldsymbol{y}_{t-1}) P_{\boldsymbol{\Theta}}(\boldsymbol{y}_t|z_t, \boldsymbol{y}_{t-1}). \quad (1)$$

Furthermore, each utterance is generated by a sequential process whereby each token $y_{t,n}$ is conditioned on all the previously generated tokens $\boldsymbol{y}_{t,<n}$, as well as the external conditioning context consisting of the dialogue act z_t and the previous turn's utterance $\boldsymbol{y}_{t-1}$, i.e.,

$$P_{\boldsymbol{\Theta}}(\boldsymbol{y}_t|z_t, \boldsymbol{y}_{t-1}) = \prod_{n=1}^{|\boldsymbol{y}_t|} P_{\boldsymbol{\Theta}}(y_{t,n}|\boldsymbol{y}_{t,<n}, z_t, \boldsymbol{y}_{t-1}). \quad (2)$$

Importantly, the decomposition of the joint distribution in Equation 1 allows dynamic programming for exact decoding (§2.2). One possible extension of our framework is to investigate a higher-order Markov model, although one needs to be conscious about the trade-off between the increase in the computational complexity of training/decoding with higher-order Markov models versus the potential gain in classification quality.

We now turn our attention to the neural architecture used to realise the components of our probabilistic model (Figure 1). We define the neural model for the conditional probability of the next dialogue act as follows:

$$P_{\boldsymbol{\Theta}}(z_t|z_{t-1}, \boldsymbol{y}_{t-1}) = \text{softmax}(\boldsymbol{W}_{cz}^{(z_{t-1})}\boldsymbol{c}_t + \boldsymbol{b}_z^{(z_{t-1})}), \quad (3)$$

where $\boldsymbol{c}_t$ is the *context* vector summarising the information from the previous utterance $\boldsymbol{y}_{t-1}$, and $\boldsymbol{W}_{cz}^{(z_{t-1})}$ and $\boldsymbol{b}_z^{(z_{t-1})}$ are the softmax parameter *gated* on the previous dialogue act z_{t-1}. Due to gating, the number of parameters of the model may increase significantly; therefore, we have also explored a variant where only the bias term $\boldsymbol{b}_z^{(z_{t-1})}$ is gated. We define the neural model for generating the tokens of the current utterance as follows:

$$P_{\boldsymbol{\Theta}}(y_{t,n}|\boldsymbol{y}_{t,<n}, z_t, \boldsymbol{y}_{t-1}) = \text{softmax}(\boldsymbol{W}_{hy}^{(z_t)}\boldsymbol{h}_{t,n-1} + \boldsymbol{W}_c\boldsymbol{c}_t + \boldsymbol{b}_y), \quad (4)$$

where the weight matrix $\boldsymbol{W}_{hy}^{(z_t)}$ is gated based on z_t, $\boldsymbol{c}_t$ summarises the previous utterance, and $\boldsymbol{h}_{t,n-1}$ is the state of an utterance-level RNN summarising all the previously generated tokens:

$$\boldsymbol{h}_{t,n-1} = \boldsymbol{f}(\boldsymbol{h}_{t,n-2}, \boldsymbol{E}_{y_{t,n-1}}), \quad (5)$$

where $\boldsymbol{E}_{y_{t,n-1}}$ provides the embedding of the token $y_{t,n-1}$ from the embedding table $\boldsymbol{E}$, and $\boldsymbol{f}$ can be any non-linear function, i.e., the simple *sigmoid* applied to elements of a vector, or the more complex Long-Short-Term-Memory unit (*LSTM*) (Graves, 2013; Hochreiter and Schmidhuber, 1997), or the Gated-Recurrent-Unit (*GRU*) (Chung et al., 2014; Cho et al., 2014).

In what follows, we elaborate on how to best summarise the information from the previous utterance in $\boldsymbol{c}_t$, and how to decode for the best sequence of dialogue acts given a trend model.

2.1 The Gated Attention Mechanism

Given a sequence of words in an utterance $\{y_1, \ldots, y_n\}$, we would like to compress its information in $\boldsymbol{c}$, which is then used in the conditioning contexts of other components of the model. Typically, the last hidden state of the utterance-level RNN is taken to be the summary vector: $\boldsymbol{c} = \boldsymbol{h}_n$. However, it has been shown that *attending* to all RNN states is more effective.

The traditional attention mechanism (Bahdanau et al., 2014) employs a probability vector $\boldsymbol{a}$ over the words of the input utterance to summarise it. The attention elements in $\boldsymbol{a}$ are typically calculated from the current input y_n, and the previous hidden state $\boldsymbol{h}_{n-1}$:

$$\alpha_n = g(\boldsymbol{h}_{n-1}, \boldsymbol{E}_{y_n}) \quad , \quad a_n = \frac{e^{\alpha_n}}{\sum_{n'=1}^{n} e^{\alpha_{n'}}},$$

where g is a non-linear function. Once the attention is defined, the representation of the input is constructed as

$$\boldsymbol{c} = \sum_n a_n \boldsymbol{h}_n. \tag{6}$$

The problem with this traditional attention model is that the final hidden state is a function of all the inputs, hence it is usually more "informative" than the earlier hidden states due to semantic accumulation (Wang et al., 2016). Thus, most of the attention signal is assigned to the hidden states toward the end of a sequence. In DA classification, this may not be desirable, since an important token with respect to a dialogue act can appear anywhere in an utterance. We call this the *attention bias* problem.

We propose a novel gated attention mechanism, which is inspired by the gating mechanism in LSTMs, to fix the *attention bias* problem. Similar to the forget gate of LSTMs, we use the available information to calculate an attention gate that learns whether to allow the whole *input signal* to pass through or to forget all or a part of the input signal:

$$\boldsymbol{a}_n = \boldsymbol{g}(\boldsymbol{h}_{n-1}, \boldsymbol{E}_{y_n}) \tag{7}$$

$$\boldsymbol{x}_n = \boldsymbol{a}_n \odot \boldsymbol{E}_{y_n} \tag{8}$$

$$\boldsymbol{h}_n = \boldsymbol{f}(\boldsymbol{h}_{n-1}, \boldsymbol{x}_n) \tag{9}$$

where $\odot$ represents element-wise multiplication.

After filtering the important signal from the input token, the information from our tokens is accumulated in the last hidden state of the RNN, which we take as the summary vector $\boldsymbol{c} = \boldsymbol{h}_n$. Note that since the gated attention is applied to the input before the RNN calculations, it is not affected by the attention bias.

2.2 Inference: Viterbi Decoding

For prediction, we choose the sequence of dialogue acts with the highest posterior probability:

$$\arg\max_{\boldsymbol{z}'_{1:T}} P_{\boldsymbol{\Theta}}(\boldsymbol{z}'_{1:T} | \boldsymbol{y}_{1:T}) = \arg\max_{\boldsymbol{z}'_{1:T}} P_{\boldsymbol{\Theta}}(\boldsymbol{z}'_{1:T}, \boldsymbol{y}_{1:T})$$

Since the joint probability is decomposed further according to Equation 1, we can make use of dynamic programming to find the highest probability sequence of dialogue acts. Specifically, the model endows each latent variable z_t with a unary potential $P_{\boldsymbol{\Theta}}(\boldsymbol{y}_t | z_t, \boldsymbol{y}_{t-1})$ and binary potential $P_{\boldsymbol{\Theta}}(z_t | z_{t-1}, \boldsymbol{y}_{t-1})$ functions. $P_{\boldsymbol{\Theta}}(\boldsymbol{y}_t | z_t, \boldsymbol{y}_{t-1})$ and $P_{\boldsymbol{\Theta}}(z_t | z_{t-1}, \boldsymbol{y}_{t-1})$ are akin to the *emission* and *transition* functions of an HMM, and are calculated using Equations 2 and 3 respectively. Furthermore, the model has been carefully designed so that the hidden states in the RNNs encoding the utterances to form the context vector $\boldsymbol{c}_t$ (the representation of the previous utterance) are *not* affected by the sequence of dialogue acts, which is crucial to making the inference amenable to dynamic programming. The resulting inference algorithm is akin to the Viterbi algorithm for HMMs.

3 Experiments

Datasets. We conduct our experiments on the MapTask and Switchboard corpora. The MapTask Dialog Act corpus (Anderson et al., 1991) consists of 128 conversations and more than 27000 utterances in an instruction-giving scenario. There are 13 DA types in this corpus. For the experiments, the available data is split into three parts, train/test/validation with 103, 13 and 12 conversations respectively.

The Switchboard Dialog Act corpus (Jurafsky et al., 1997) consists of 1155 transcribed telephone conversations with around 205000 utterances. In contrast with the MapTask conversations, which are task-oriented, the Switchboard corpus consists mostly of general topic conversations. The Switchboard tag set has 42 DAs.[1]

[1]The original size of the tag set for Switchboard is 226, which was then collapsed into 42

	without HMM	gate bias HMM	gate all HMM
no attn.	60.97%	64.60%	63.55%
traditional	61.72%	64.73%	65.19%
gated attn.	62.21%	**65.94%**	**65.94%**

Table 1: Comparison of our model variants on the MapTask corpus.

Baselines. On MapTask, to the best of our knowledge, there is no standard data split, thus, we make the comparison against our implementation of strong baselines such as HMM-trigram (Stolcke et al., 2000) and instance-based random forest classifier (1/2/3-gram features). Ji *et al.*'s (2016) results for this corpus are obtained by running their publicly available code with the same hyper parameters as those used by our models. We also report the results of Julia *et al.* (2010)[2] and Surendran *et al.* (2006). However, the experimental setup of these two works differs from ours, hence their results are not directly comparable to ours.

On Switchboard, we compare our results with strong baselines using the experimental setup from Kalchbrenner and Blunsom (2013) and Stolcke *et al.* (2000).[3]

Our Model Configurations. We experiment with several variants of our model to explore the effectiveness of our two improvements: the HMM-like connection and the gated attention mechanism. For the HMM connection, we consider three choices: gating all parameters (Equation 3), gating only the bias, and no connection. For the attention, we consider three choices: our new gated attention mechanism, the traditional attention, and no attention. Thus, in total, we explore nine model variants.

All the model variants are implemented with the CNN package[4] and trained with Adagrad (Duchi et al., 2011) using dropout (Srivastava et al., 2014). They share the same word-embedding size (128) and hidden vector size (64).[5]

[2]Julia *et al.* (2010) employed both text transcription and audio signal. Here, we report the results obtained with the transcription.

[3]There have been other works with different experimental setups (Gambäck et al., 2011; Webb and Ferguson, 2010) that obtained accuracies ranging from 77.85% to 80.72%. However, these results are not directly comparable to ours.

[4]https://github.com/clab/cnn-v1.

[5]The experiments were executed on an Intel Xeon E5-2667 CPU with 16GB of RAM. The training time for each MapTask model is less than a day, the training time for each Switchboard model takes up to four weeks.

Models	**Accuracy**
Julia *et al.* (2010)	55.40%
Surendran *et al.* (2006)	59.10%
HMM (Stolcke *et al.* (2000))	51.40%
Random Forest (n-gram)	55.72%
Ji *et al.* (2016)	60.97%
Our model	
gated attn. + gated HMM bias	**65.94%**
gated attn. + gated HMM all	**65.94%**

Table 2: Results on MapTask data.

Models	**Accuracy**
Stolcke *et al.* (2000)	71.0%
Kalchbrenner and Blunsom (2013)	73.9%
Ji *et al.* (2016)	72.5%
Shen and Lee (2016)	72.6%
our model	
gated attn. + gated HMM bias	**74.2%**
gated attn. + gated HMM all	74.0%

Table 3: Results on Switchboard data.

Results and Analysis. Table 1 shows the classification accuracy of the nine variants of our model on the MapTask corpus. The classification accuracy of the two best variants of our model and the baselines appears in Tables 2 and 3 for MapTask and Switchboard respectively. The bold numbers in each table show the best accuracy achieved by the systems. As seen in these tables, our best models outperform strong baselines for both corpora.[6]

Table 1 shows that adding the attention mechanism is beneficial, as the traditional attention models always outperform their non-attention counterparts. The gated attention configurations, in turn, outperform those with the traditional attention mechanism by 0.49%-1.21%. Interestingly, the accuracy of Shen and Lee's (2016) classifier, which employs an attention mechanism, is lower than that obtained by Kalchbrenner and Blunsom (2013), whose mechanism does not use attention. We believe that the difference in performance is not due to the attention mechanism being ineffective, but because Shen and Lee (2016) treat the classification of each utterance independently. In contrast, Kalchbrenner and Blunsom (2013) take

[6]Ji *et al.* (2016) reported an accuracy of 77.0% on the Switchboard corpus, but their paper does not provide enough information about the experimental setup to replicate this result (hyper-parameters, train/test/development split). Thus, we ran the paper's publicly available code with our experimental settings, and report the result in our comparison.

the sequential nature of dialog acts into account, and run an RNN across the conversation, which conditions the generation of a dialogue act on the dialogue acts and utterances in all the previous dialogue turns.

As seen in Table 1, the performance gain from the HMM connection is larger than the gain from the attention mechanism. Without the attention mechanism, the HMM connection brings an increase of 3.63% with the gated bias HMM configuration and 2.58% with the fully gated HMM configuration. With the use of traditional attention, the improvement is 3.01% for the bias HMM configuration and 3.47% for the gated HMM configuration. Finally with the gated attention in place, the two HMM configurations improve the accuracy by 3.73%.

We used McNemar's test to determine the statistical significance between the predictions of different models, and found that our model with both innovations (HMM connections and gated attention) is statistically significantly better than the variant without these innovations with $\alpha < 0.01$.

4 Conclusions

In this work, we have proposed a new gated attention mechanism and a novel HMM-like connection in a generative model of utterances and dialogue acts. Our experiments show that these two innovations significantly improve the accuracy of DA classification on the MapTask and Switchboard corpora. In the future, we plan to apply these two innovations to other sequence-to-sequence learning tasks. Furthermore, DA classification itself can be seen as a preprocessing step in a dialogue system's pipeline. Thus, we also plan to investigate the effect of improvements in DA classification on the downstream components of a dialogue system.

References

Anne H Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, et al. 1991. The HCRC map task 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* .

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* .

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* .

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.

Björn Gambäck, Fredrik Olsson, and Oscar Täckström. 2011. Active learning for dialogue act classification. In *Interspeech 2011 – Proceedings of the International Conference on Spoken Language Processing*. pages 1329–1332.

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.

Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. 2016. A latent variable recurrent neural network for discourse-driven language models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 332–342. http://www.aclweb.org/anthology/N16-1037.

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(04):289–311.

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

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 *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pages 1412–1421. http://aclweb.org/anthology/D15-1166.

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. http://dl.acm.org/citation.cfm?id=2627435.2670313.

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 *Interspeech 2006 – Proceedings of the International Conference on Spoken Language Processing*. pages 1950–1953.

Bingning Wang, Kang Liu, and Jun Zhao. 2016. Inner attention based recurrent neural networks for answer selection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pages 1288–1297. http://www.aclweb.org/anthology/P16-1122.

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: Posters*. pages 1310–1317.