# Preserving Distributional Information in 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

This paper introduces a novel training/decoding strategy for sequence labeling. Instead of greedily choosing a label at each time step, and using it for the next prediction, we retain the probability distribution over the current label, and pass this distribution to the next prediction. This approach allows us to avoid the effect of label bias and error propagation in sequence learning/decoding. Our experiments on dialogue act classification demonstrate the effectiveness of this approach. Even though our underlying neural network model is relatively simple, it outperforms more complex neural models, achieving state-of-the-art results on the MapTask and Switchboard corpora.

## 1 Introduction

Dialogue Act (DA) classification is a sequence-labeling task, where a sequence of utterances is mapped into a sequence of DAs. The DAs are semantic classifications of the utterances, and different corpora usually have their own DA labels.

Two of the most popular DA classification datasets are Switchboard (Godfrey et al., 1992; Jurafsky et al., 1997) and MapTask (Anderson et al., 1991). There have been many works on DA classification applied to these two datasets; some focus on textual data (Kalchbrenner and Blunsom, 2013; Stolcke et al., 2000), while others explore speech data (Julia et al., 2010). The classification methods used can be broadly divided into *instance-based methods* (Julia et al., 2010; Gambäck et al., 2011) and *sequence-labeling methods* (Stolcke et al., 2000; Kalchbrenner and Blunsom, 2013; Ji et al., 2016; Shen and Lee, 2016; Tran et al., 2017). Instance-based methods treat each utterance as an independent data point, which allows the application of general machine learning models, such as Support Vector Machines. Sequence-labeling methods include methods based on Hidden Markov Models (HMMs) (Stolcke et al., 2000) and neural networks (Kalchbrenner and Blunsom, 2013; Ji et al., 2016; Shen and Lee, 2016; Tran et al., 2017).

Stolcke *et al.* employed an HMM, using a Language Model to produce emission probabilities. The neural models are particularly successful, posting a higher accuracy on Switchboard than the HMM. Specifically, Kalchbrenner and Blunsom (2013) model a DA sequence with a recurrent neural network (RNN) where sentence representations are constructed by means of a convolutional neural network (CNN); Ji *et al.* (2016) treat the labels as latent variables in a generative RNN; Shen and Lee (2016) employ attentional RNNs for the independent prediction of DAs; and Tran *et al.* (2017) model the DAs in a conversation by means of a hierarchical RNN. In this paper, we also rely on RNNs, but our architecture is much simpler than the above neural models, while posting competitive results.

Most neural network models for DA classification employ greedy decoding (Tran et al., 2017; Ji et al., 2016), as its speed and simplicity support an on-line decoding process (i.e., producing a label immediately after receiving an utterance). For sequential labeling, the DA label in the current time step is very important (Tran et al., 2017). However, using a greedy approach to connect the current label directly to the next label may degrade


*Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2151–2156
Copenhagen, Denmark, September 7–11, 2017. 

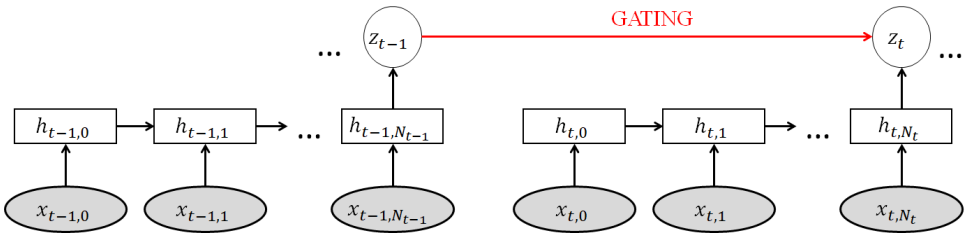


Figure 1: Model architecture.

performance, because the current predicted label may be noisy, which in turn leads to the propagation of errors through the sequence (Tran et al., 2017; Ranzato et al., 2015).

Recently, Bengio *et al.* (2015) proposed a technique called *Scheduled Sampling* that tries to solve the label-bias problem by alternating between the predicted label and the correct label during training. This makes the model gradually adapt to the noisiness of the predicted label. However, this method still relies upon a single current label, and, by omitting the distribution over the possible labels, this model loses information about the current stage. In contrast, we propose to condition the next label on a predicted distribution of the current label. Specifically, we introduce two variants of this idea: the *Uncertainty Propagation* model and the *Average Embedding* model.

## 2 Sequential DA Prediction

We are interested in predicting DAs $\{z_1, \ldots, z_t\}$ in a conversation as we receive textual utterances $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$ sequentially. Importantly, we do not have access to future utterances when predicting a DA at time $t$.

**Model.** We propose a discriminative model, where the probability of DAs conditioned on utterances is decomposed as follows (Figure 1):

$$p(\boldsymbol{z}_{1:t}|\boldsymbol{x}_{1:t}) = \prod_{i=1}^{t} p_{\boldsymbol{\theta}}(z_i|z_{i-1}, \boldsymbol{x}_i) \qquad (1)$$

where $\boldsymbol{z}_{1:t}$ and $\boldsymbol{x}_{1:t}$ respectively denote the sequence of DAs and utterances up to time step $t$. Our model resembles a maximum entropy Markov model, as it conditions the label of the next time step on the label of the current step and the next received utterance. The conditional distribution term $p_{\boldsymbol{\theta}}(z_i|z_{i-1}, \boldsymbol{x}_i)$ is realised by neural models as follows:

$$z_i|z_{i-1}, \boldsymbol{x}_i \sim softmax(\boldsymbol{W}^{(z_{i-1})}\boldsymbol{c}(\boldsymbol{x}_i) + \boldsymbol{b}^{(z_{i-1})}) \qquad (2)$$

where $\boldsymbol{c}(\boldsymbol{x}_i)$ is the distributed representation of utterance $\boldsymbol{x}_i$ (discussed below), and $\{\boldsymbol{W}^{(z_{i-1})}, \boldsymbol{b}^{(z_{i-1})}\}$ are DA-specific parameters gated on the current DA $z_{i-1}$.

The encoding function for an utterance is an RNN with long-short term memory (LSTM) units (Graves, 2013; Hochreiter and Schmidhuber, 1997), where the final hidden state of the RNN is taken as the representation of the whole sequence of text:

$$\boldsymbol{h}_{t,n} = \boldsymbol{f}_{\boldsymbol{\phi}}(\boldsymbol{h}_{t,n-1}, \boldsymbol{e}(x_{t,n})) \;\;,\;\; \boldsymbol{c}(\boldsymbol{x}_t) = \boldsymbol{h}_{t,N_t} \qquad (3)$$

where $x_{t,n}$ is the $n$-th token in the $t$-th utterance, and $N_t$ is the length of the utterance.

The parameter set of our model $\boldsymbol{\theta}$ includes $\{\boldsymbol{W}^{(\ell)}, \boldsymbol{b}^{(\ell)}\}_{\ell=1}^{L}$ for the gating component (where $L$ is the number of DAs), as well as the LSTM parameters $\boldsymbol{\phi}$ and the word-embedding table $\{\boldsymbol{e}(w)\}_{w \in \mathcal{W}}$, where $\mathcal{W}$ is the dictionary.

**Uncertainty Propagation.** In this model, the distribution over the labels at the current time step is passed to the next time step. Specifically, the quantity of interest is the posterior probability of the DA of the next time step given all the utterances observed so far. This posterior probability can be rewritten as

$$\begin{aligned} p_{\boldsymbol{\theta}}(z_t|\boldsymbol{x}_{1:t}) &= \sum_{z_1,\ldots,z_{t-1}} p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:t}|\boldsymbol{x}_{1:t}) \\ &= \sum_{z_{t-1}} p_{\boldsymbol{\theta}}(z_t|z_{t-1}, \boldsymbol{x}_t) p_{\boldsymbol{\theta}}(z_{t-1}|\boldsymbol{x}_{1:t-1}) \end{aligned} \qquad (4)$$

According to Equation 4, the label uncertainty at the next time step $t$ can be computed by a dynamic programming algorithm based on the label uncertainty of the current time step combined with the local potentials $p_{\boldsymbol{\theta}}(z_t|z_{t-1}, \boldsymbol{x}_t)$.

The use of posterior probability for prediction is also motivated by the minimum Bayes risk decoding (MBR). In the sequential setting, we are interested in predicting the next DA that minimizes the expected loss

$$\begin{aligned} &\arg\min_{\hat{z}_t} \sum_{z_1,\ldots,z_{t-1}} p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:t}|\boldsymbol{x}_{1:t}) loss(z_t, \hat{z}_t) \\ = \;& \arg\min_{\hat{z}_t} p_{\boldsymbol{\theta}}(z_t|\boldsymbol{x}_{1:t}) loss(z_t, \hat{z}_t) \end{aligned} \quad (5)$$

where $\hat{z}_t$ is the predicted label, $z_t$ is the actual label, and $loss(z_t, \hat{z}_t) = 1_{z_t \neq \hat{z}_t}$.

In addition to decoding, we use posterior probability when training the model. That is, our training objective is

$$\sum_{(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}) \in \mathcal{D}} \sum_{t=1}^{T} \log p_{\boldsymbol{\theta}}(z_t|\boldsymbol{x}_{1:t}) \quad (6)$$

where $\mathcal{D}$ is the set of conversations in the training set, each consisting of a sequence of utterances $\boldsymbol{x}_{1:T}$ annotated with its gold sequence of DAs $\boldsymbol{z}_{1:T}$.

**Average Embedding.** This model offers a new perspective where a neural net combines an inference machine and a model (rather than simply encoding a model). Specifically, this model represents in its architecture, through a weighted sum of embeddings, the inference procedure encoded in Equation 4 for the Uncertainty Propagation model:

$$softmax(E_{q(z_{t-1})}[\boldsymbol{W}^{(z_{t-1})}]\boldsymbol{c}(\boldsymbol{x}_t) + E_{q(z_{t-1})}[\boldsymbol{b}^{(z_{t-1})}]) \quad (7)$$

where $q(z_t)$ is an embedding that represents the uncertainty at time step $t$. $q(z_t)$ is computed sequentially as new utterances are received, and used in both decoding and training.

This formulation contrasts with Uncertainty Propagation, where the expectation is over the distributions:

$$E_{p_{\boldsymbol{\theta}}(z_{t-1}|\boldsymbol{x}_{1:t-1})}[softmax(\boldsymbol{W}^{(z_{t-1})}\boldsymbol{c}(\boldsymbol{x}_t) + \boldsymbol{b}^{(z_{t-1})})] \quad (8)$$

It is worth noting that Equations 7 and 8 yield the same result if the distributions involved in calculating the expectations are point-mass distributions and they are equal.

Although we could have used a more elaborate neural architecture as the inference machine for the Average Embedding model, we employed a simple *softmax* architecture to make this model comparable with the principled inference algorithm for our Uncertainty Propagation model, which is based on Equation 4.

**Comparison to traditional graphical models** Our models have several similarities with the traditional HMM model and inference algorithms, such as Forward-Backward decoding and the Viterbi algorithm. However, there are some key differences. Firstly, our model is discriminative, whereas HMM is generative. Secondly, our method is designed for online decoding (the future inputs to a specific classification decision are unknown), whereas both Forward-Backward decoding and Viterbi require access to the whole sequence. Thirdly, Viterbi's objective is to decode for the most probable sequence of labels, whereas our decoding algorithm's objective is to find the sequence of most probable labels (conditioned on the inputs observed so far). Lastly, our Uncertainty Propagation model is not only a basis for decoding, but also for training (the training objective in Equation 6 requires the calculation of the posterior probability in Equation 4). Overall, the best analogue of our Uncertainty Propagation model to methods used in HMMs and other graphical models is the forward message calculation in the Forward-Backward algorithm.

## 3 Experiments

### 3.1 Data sets

For our experiments, we use the MapTask and Switchboard corpora.

The MapTask Dialog Act corpus (Anderson et al., 1991) consists of 128 conversations tagged with 13 DAs. The MapTask conversations focus on instructions and clarifications — in the MapTask experiment, there is one instruction giver and one instruction follower. The task of the instruction giver is to guide the instruction follower to follow a pre-determined path, and the instruction follower must draw this path on his/her map. We use 12 conversations for validation, 13 for testing, and the rest for training.

The Switchboard Dialog Act corpus (Godfrey et al., 1992; Jurafsky et al., 1997) consists of 1155 transcribed telephone conversations about general topics, encoded into 42 DAs. We use the experimental setup proposed by Stolcke *et al.* (2000): 1115 conversations for training and 19 for testing.

### 3.2 Baselines

Our first baseline is the model without any current label information. Next, we compare our models with other strategies for incorporating the current

| Models | Accuracy Switchboard | MapTask |
|---|---|---|
| No current label | 72.93% | 61.27% |
| True current label | 73.15% | 63.36% |
| Predicted current label | 73.91% | 64.53% |
| Scheduled Sampling | 74.43% | 64.50% |
| Average Embedding | 75.04% | 65.09% |
| Uncertainty Propagation | **75.61**% | **65.87**% |

Table 1: Results of different strategies to leverage the current label.

labels, viz those that use *predicted label* in training, and those that use *correct label*. These models simply employ the predicted/correct label to gate the parameters in Equation 2 during training. During testing, both models can only use the predicted label.

Another baseline is Bengio *et al.*'s (2015) Scheduled Sampling technique, where the training model uses the current correct label with probability $p$ and the predicted label with probability $1 - p$, and $p$ is scheduled to decrease over time. This strategy tries to solve the label-bias problem by making the model gradually adapt to the noisy predicted current label.

Finally, we consider the results obtained by corpus-specific baselines, viz (Julia et al., 2010; Surendran and Levow, 2006; Tran et al., 2017) for MapTask, and (Stolcke et al., 2000; Kalchbrenner and Blunsom, 2013; Ji et al., 2016; Shen and Lee, 2016; Tran et al., 2017) for Switchboard.

### 3.3 Results

Table 1 compares our results with those obtained by the baselines. Our two models, Uncertainty Propagation and Average Embedding, outperform all the baselines. Among these two models, Uncertainty Propagation, which is more analytically grounded, outperforms the Average Embedding model. Using the true current label during training seems to degrade performance compared to using the predicted label, which is expected, since the true label is not available during testing. The Scheduled Sampling method performs similarly to the predicted-label method for the MapTask corpus, and outperforms this method for the Switchboard corpus.

Tables 2 and 3 compare our models' performance on the MapTask and Switchboard corpora respectively with that of several strong baselines. On MapTask, we achieved the best results for

| Baseline models | Accuracy |
|---|---|
| Julia *et al.* (2010) | 55.4 % |
| Surendran and Levow (2006) | 59.1% |
| Tran *et al.* (2017) | 61.6% |
| **Our models:** | |
| Average Embedding | 62.6% |
| Uncertainty Propagation | **62.9**% |

Table 2: Results on MapTask data.

| Baseline models | Accuracy |
|---|---|
| Stolcke *et al.* (2000) | 71.0% |
| Shen and Lee (2016) | 72.6% |
| Kalchbrenner and Blunsom (2013) | 73.9% |
| Tran *et al.* (2017) | 74.5% |
| Ji *et al.* (2016) | (77.0%) 72.5% |
| **Our models:** | |
| Average Embedding | 75.0% |
| Uncertainty Propagation | **75.6**% |

Table 3: Results on Switchboard data.

textual input, using the four-fold cross-validation setup used by Surendran and Levow (2006) and Julia *et al.* (2010). On Switchboard, we also obtained the best results among the systems with the same experimental setting. It is worth noting that Ji *et al.* (2016) reported a higher accuracy of 77.0%, but the paper does not provide enough information about the experimental setup to replicate this result, and we only got 72.5% accuracy using the paper's publicly available code.

### 3.4 Analysis

To quantify the effectiveness of the different models on reducing the label-bias problem, we calculate the probability of the models making a correct prediction after they have made a sequence of $n$ mistakes. We expect our models, Uncertainty Propagation and Average Embedding, to be more robust than the label-sensitive baselines in recovering from errors.

The results in Table 4 confirm our expectations. The simple model with no current label, while performing worse than all other models in accuracy, does not suffer from the label-bias problem. Among the models with current label information, Uncertainty Propagation suffers the least from label bias. It even outperforms the model with no current label on Switchboard for all values of $n$, and on MapTask for $n = 2$. Interestingly, Average Embedding performs quite well for $n = 1$, but

| | MapTask | | | Switchboard | | |
|---|---|---|---|---|---|---|
| | $n=1$ | $n=2$ | $n=3$ | $n=1$ | $n=2$ | $n=3$ |
| **Not affected by label bias:** | | | | | | |
| No Previous label | 60.29% | 56.90% | 55.67% | 66.99% | 63.10% | 56.52% |
| **Affected by label bias:** | | | | | | |
| True current label | 53.12% | 50.38% | 47.89% | 61.74% | 60.93% | 60.71% |
| Predicted current label | 55.89% | 53.65% | 49.10% | 64.38% | 62.21% | 62.59% |
| Scheduled Sampling | 54.28% | 53.32% | 50.00% | 64.67% | 63.49% | 60.87% |
| Average Embedding | 56.50% | 53.76% | 52.56% | 66.51% | 61.71% | 55.22% |
| Uncertainty Propagation | **57.13%** | **57.37%** | **53.93%** | **67.78%** | **66.57%** | **66.36%** |

Table 4: Probability that the models recover from a sequence of $n$ prediction mistakes.

its ability to recover from errors drops quickly as the length of the erroneous conditioning sequence increases, especially on Switchboard, where the number of labels is higher. This may explain its slightly lower accuracy compared to the Uncertainty Propagation model. However, in general, the difference in accuracy between these two models is small, because they are rather unlikely to make several consecutive errors.

## 4 Conclusion

In this paper, we proposed two strategies to encode current label uncertainty in sequence-labeling RNN models. The experimental results show that our models achieve a very strong performance on the MapTask and Switchboard corpora using a simple underlying RNN architecture.

Although we experimented with DA classification, the idea presented in this paper is general, and can be applied to many sequence-labeling tasks. Our approach is particularly suitable for tasks involving streaming data where the model only has access to current and previous observations.

In the future, we plan to combine our strategies with more complex neural architectures, and explore their application to other sequence-labeling problems.

## 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 MapTask corpus. *Language and speech* 34(4):351–366.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*. pages 1171–1179.

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.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*. IEEE, volume 1, pages 517–520.

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*. San Diego, California, 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* .

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* .

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.

Quan Hung Tran, Ingrid Zukerman, and Gholamreza Haffari. 2017. A hierarchical neural model for learning sequences of dialogue acts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain, pages 428–437. http://www.aclweb.org/anthology/E17-1041.