Dynamic Memory Networks for Dialogue Topic Tracking

Seokhwan Kim
Adobe Research
San Jose, CA, USA
seokim@adobe.com

Abstract

Dialogue topic tracking aims at identifying the topic states in on-going multi-topic conversations. This paper proposes dynamic memory networks for dialogue topic tracking to learn the context states of conversations represented by multiple memory slots. The slot values are managed by the gated recurrent architectures with the update and reset gates considering cross-slot interactions. The experimental results show that our proposed models significantly improved both the sequential labelling and segmentation performances in topic tracking of human-human conversations in comparison to the other neural network baselines.

Introduction

Recent advancements in artificial intelligence have contributed to closing the gap between the technologies and their uses in our daily life. One of the practical successes is that natural language dialogues have been used as a means of human machine interface implemented in many consumer devices. However, the current dialogue systems still have limited capabilities of conducting a coherent conversation across multiple different topics, which is generally taken for granted in human conversations.

Dialogue topic tracking is a sub-task of dialogue state tracking focusing on the topic-related states in an on-going multi-topic conversation. While many previous studies on multi-topic dialogue processing aimed at directly building dialogue system components for topic categorization (Lin, Wang, and Lee 1999; Nakata, Ando, and Okumura 2002; Lagus and Kuusisto 2002; Adams and Martell 2008; Ikeda et al. 2008; Celikyilmaz, Hakkani-Tür, and Tür 2011) or dialogue flow management (Bohus and Rudnicky 2003; Roy and Subramaniam 2006; Lee, Jung, and Lee 2008) in human-machine conversations, recent studies (Morchid et al. 2014a; 2014b; Esteve et al. 2015; Kim, Banchs, and Li 2016) have addressed this problem on human-human conversations as part of the efforts in understanding human behaviors in dealing with multiple topics.

In our prior work (Kim, Banchs, and Li 2016), dialogue topic tracking was formulated as an utterance-level sequential labelling problem and proposed various neural network architectures including convolutional and recurrent neural networks on it. We reported that the local features captured by the convolutional architectures led to significant improvements of the topic tracking performances. On the other hand, the temporal contexts modelled by the recurrent networks only showed a marginal effect.

In this paper, we propose dynamic memory networks for dialogue topic tracking towards better representation of dialogue contexts compared to the neural network architectures in the prior work. Our models represent the latent dialogue state at each given time step as a set of read-writable memory slots, inspired by the neural memory models (Graves, Wayne, and Danihelka 2014; Graves et al. 2016; Henaff et al. 2016). Each memory slot is updated through a given dialogue sequence by the content-based operations in gated recurrent networks.

Unlike the single gating mechanism in the previous memory networks, we propose an additional reset gate to explicitly filter out any outdated context from the state representation. Additionally, the cross-slot interactions are newly introduced into both the update and reset gates in order to overcome the limitations of the distributed architectures due to the lack of consideration of any correlation between different memory slots.

In the remainder of this paper, we present a problem definition of dialog topic tracking and describe our dynamic memory network models for this problem. Then, the evaluation results of the models are reported followed by the conclusions.

Dialogue Topic Tracking

Following our prior work (Kim, Banchs, and Li 2016), we define dialogue topic tracking as a multi-class classification problem at each time step in a given dialogue to predict the label encoded in B/I/O tagging scheme (Ramshaw and Marcus 1995) as follows:

$$f(t) = \begin{cases} \text{B-}\{c \in C\} & \text{if } u_t \text{ is at the beginning} \\ & \text{of a segment belongs to } c, \\ \text{I-}\{c \in C\} & \text{else if } u_t \text{ is inside a} \\ & \text{segment belongs to } c, \\ \text{O} & \text{otherwise,} \end{cases} \quad (1)$$

where u_t is the utterance at the t-th time step in a given dialogue session and C is a closed set of topic categories.

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

t	Speaker	Utterance (u_t)	$f(t)$
1	Guide	How can I help you?	B-OPEN
2	Tourist	Can you recommend some good places to visit in Singapore?	B-ATTR
3	Guide	Well if you like to visit an icon of Singapore, Merlion will be a nice place to visit.	I-ATTR
4	Tourist	Okay. But I'm particularly interested in amusement parks.	B-ATTR
5	Guide	Then, what about Universal Studio?	I-ATTR
6	Tourist	Good! How can I get there from Orchard Road by public transportation?	B-TRSP
7	Guide	You can take the red line train from Orchard and transfer to the purple line at Dhoby Ghaut. Then, you could reach HarbourFront where Sentosa Express departs.	I-TRSP
8	Tourist	How long does it take in total?	I-TRSP
9	Guide	It'll take around half an hour.	I-TRSP
10	Tourist	Alright.	I-TRSP
11	Guide	You could spend a whole afternoon at the park by its closing time at 6pm.	B-ATTR
12	Tourist	Sounds good!	I-ATTR
13	Guide	Then, I recommend you enjoy dinner at the riverside on the way back.	B-FOOD
14	Tourist	What do you recommend there?	I-FOOD
15	Guide	If you like spicy foods, you must try chili crab which is one of our favorite dishes.	I-FOOD
16	Tourist	Great! I'll try that.	I-FOOD

Figure 1: Examples of dialogue topic tracking on a tour guide dialogue scenario. Each label is composed of a B/I/O tag followed by a topic category including ATTR, TRSP and FOOD which denotes attraction, transportation, and food, respectively.

Figure 1 shows an example dialogue session on tour guide domain with the topic tracking labels. The labels preceded by 'B-' tags indicate the segmentation boundaries at $t = \{2, 4, 6, 11, 13\}$, which divide this conversation into six segments. At the same time, the other part of the labels shows the topic category of each segment. Dialogue topic tracking aims at detecting the transitions not only across different topic categories, but also between distinguishable subjects from each other within a single category. The boundary at $t = 4$ in the above conversation is an example of intra-categorical segmentation. There is no change of topic category between before and after of it, but two adjacent segments discuss the different subjects in detail from each other.

Since we target to eventually develop a real-time topic tracker to do the predictions on an on-going dialogue, we assume that no future information is available at each time step. Thus, two keys to success in this task are: how to capture the important expressions from the current utterance; and how to incorporate the contextual features from dialogue history into the classifier. If every utterance includes any clear signal in either initiating a new topic or maintaining the existing one, the problem can be solved just by a sentence classification on the current utterance only.

However, a certain portion of human conversation has no explicit mention directly specifying any particular topic. For example, the utterances at $t = \{10, 12, 16\}$ in the above dialogue example contain very little information within each sentence itself to be used in predicting the topic labels. The other example at $t = 14$ seems to have more cues compared to the former ones, but it is also not enough to resolve its ambiguity only with this sole utterance, because this kind of expression of asking for recommendation with no specific object can belong to almost every topic category in the tour guide domain. In these cases, the dialogue contexts from previous history are expected to act as decisive features in the classifier, which is where this work mainly focuses on.

Models

The classifier f can be trained using supervised machine learning techniques, given a set of dialogues annotated with the gold standard labels. Unlike most other sentence classification tasks, this model should take the features not only from the current utterance, but also from some previous utterances in the dialogue history into account when determining the label at each time step.

Earlier studies (Kim, Banchs, and Li 2014; 2016) have shown the effectiveness of the combined features extracted from the following three different sources: the current utterance u_t, the previous utterance u_{t-1}, and the other utterances within a history window. Extracting the first two feature sets from u_t and u_{t-1} is relatively obvious as long as the feature functions for a single utterance are clearly defined. Whereas there has been no general solution yet to get the proper history features which are commonly applicable to every situation in a given conversation.

Although the previous work also includes some mechanisms to select the window size or the decay factor optimized across the whole development dataset, these static parameters are insufficient to handle various aspects changed from time to time even in the same conversation. For example, there is a gap between $t = 10$ and 12 in Figure 1 in terms of the number of previous time steps that belongs to the same segment as the current utterance, which directly affects to the optimum size of the history features in each case.

In this work, we propose neural network architectures to overcome the limitations of the previous static networks for dialogue topic tracking. In the remainder of this section, we describe and compare the models focusing on the capabilities of learning the temporal dynamics in dialogue flows.

Convolutional Neural Networks

Following the successes of convolutional neural networks (CNNs) in many natural language processing tasks (Collobert et al. 2011; Shen et al. 2014; Kalchbrenner, Grefenstette, and Blunsom 2014; Kim 2014; Lee and Dernoncourt 2016), we previously showed the significant improvements also in the dialogue topic tracking performances achieved by a CNN architecture and its variants (Kim, Banchs, and Li 2016), which were originally based on the work by Collobert et al. (2011) and Kim (2014) for sentence classification tasks.

This architecture represents a sentence of n words as an $n \times k$ matrix by concatenating the vectors each of which

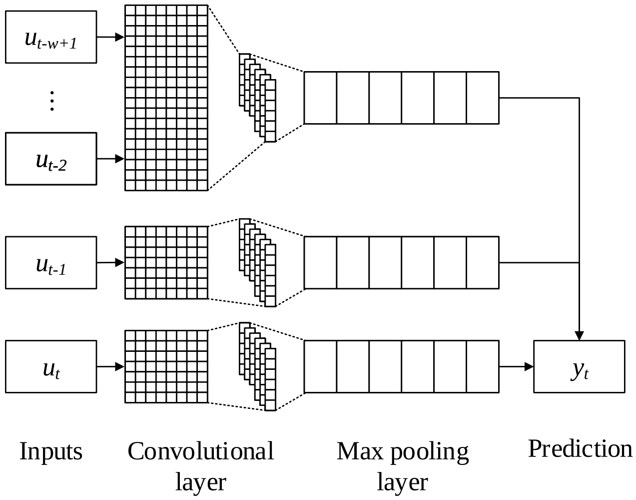


Figure 2: Convolutional neural network architecture for dialogue topic tracking.

is the k-dimensional word embedding $\vec{x}_i \in \mathbb{R}^k$ representing the i-th word in the sentence. Then, a filter $\mathcal{F} \in \mathbb{R}^{k \times m}$ with the same width k as the input matrix and a given height m generates the following convolutional feature at the i-th position:

$$c_i = \sigma\left(\mathcal{F} \cdot \vec{x}_{i:i+m-1} + b\right), \quad (2)$$

where $\vec{x}_{i:j}$ is the sub-region from the i-th row to the j-th row in the input, $b \in \mathbb{R}$ is a bias term, and σ is a non-linear activation function such as rectified linear units. A series of convolution operations sliding over from the first row to the $(n-m+1)$-th row of the input matrix produces a feature map $\vec{c} = [c_1 \cdots c_{n-m+1}] \in \mathbb{R}^{n-m+1}$ for the filter $\mathcal{F}$. Then, the maximum element $c' = \max(\vec{c})$ is selected from each feature map considered as the most important feature for the particular filter in the max-pooling layer.

The CNN model for dialogue topic tracking (Figure 2) takes an input instance at a given time step t with the following three sentences: u_t, u_{t-1}, and $u_{t-w+1:t-2}$ which denote the current utterance, the previous utterance, and the concatenation of the other utterances within w time step in the dialogue history, respectively. Both the convolution and the max-pooling operations are applied to each individual sentence separately from the others. But the same filters are commonly used in the convolutional layers across all three sentences. Finally, all the feature values are combined together and forwarded to the fully-connected layer with softmax for predicting the topic label y_t.

The feature extraction from u_t and u_{t-1} by this CNN architecture conceptually makes a good fit to one of the main objectives of this task which is capturing the key expressions determining topic states, since the importance of each cue is generally invariant to where it is originally located in a given sentence. The same approach for $u_{t-w+1:t-2}$, on the other hand, has a high chance of acting as a bottleneck in modeling dialogue histories properly. Because most contextual and temporal information is lost at the very beginning of the whole procedure by concatenating all the history utterances into the flat input representation, which is inevitable and irreversible caused by the nature of this static architecture.

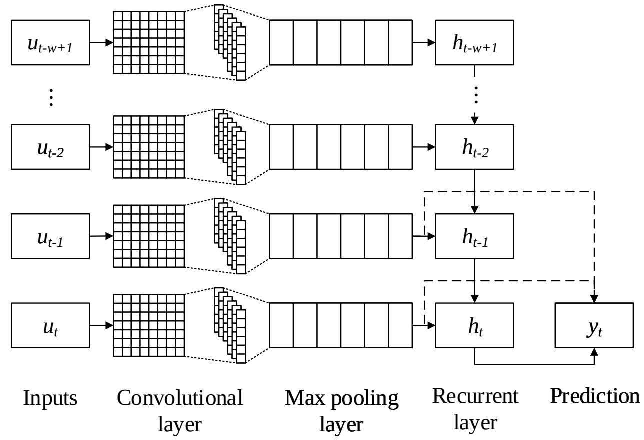


Figure 3: Recurrent convolutional neural network architecture for dialogue topic tracking.

Recurrent Convolutional Networks

To make use of the sequential dependencies between utterances in learning dialogue contexts, we propose a recurrent convolutional neural network (RCNN) architecture for dialogue topic tracking. This model (Figure 3) also takes the utterance sequence from u_{t-w+1} to u_t as the input at the time step t. As in the CNN model, each utterance u_i in the sequence is individually represented by the l-dimensional vector $\vec{u}_i \in \mathbb{R}^l$ generated through the embedding, convolution, and max pooling operations, where l is the total number of filters in the convolutional layer. Then, the vectors from $\vec{u}_{t-w+1}$ to $\vec{u}_t$ are connected in a recurrent neural network (RNN) layer, where temporal contexts are learned by recurrent computations applied to every time step in chronological order.

When a vanilla RNN unit constitutes the recurrent layer, its hidden states are updated by the operation $h_i = \sigma(W\vec{u}_i + Uh_{i-1})$, where h_i is the state at the i-th time step, and W and U are the trainable parameters which are shared across all the time steps. For better handling of any potential long-term dependencies in dialogue sequences, a gated architecture such as the long short-term memories (LSTMs) (Hochreiter and Schmidhuber 1997) and gated recurrent units (GRU) (Cho et al. 2014) may be preferred rather than the vanilla unit. Our gated implementation of RCNN uses GRU which has the following state updating mechanism:

$$h_i = z_i \circ h_{i-1} + (1 - z_i) \circ \tilde{h}_i, \quad (3)$$

$$\tilde{h}_i = \tanh\left(W_h \vec{u}_i + U_h\left(r_i \circ h_{i-1}\right)\right), \quad (4)$$

where z_i and r_i are the signal vectors for the update and reset gates, respectively, which are defined as follows:

$$z_i = \sigma\left(W_z \vec{u}_i + U_z h_{i-1}\right), \quad (5)$$

$$r_i = \sigma\left(W_r \vec{u}_i + U_r h_{i-1}\right). \quad (6)$$

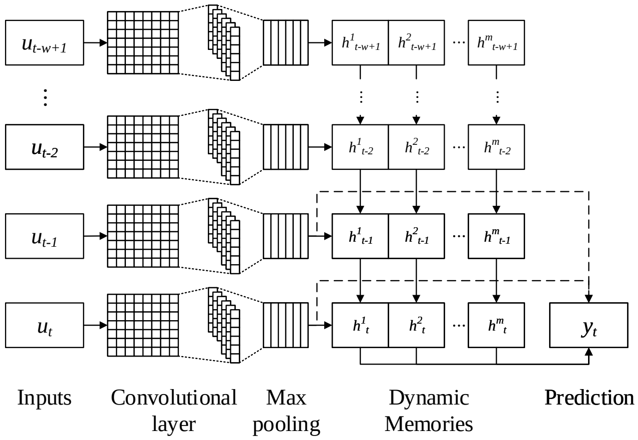


Figure 4: Dynamic memory network architecture for dialogue topic tracking.

Intuitively, the update gate determines how much of the previous information is reflected in the current state and the reset gate indicates what part of the previous state is relevant to the updated state. Both gates share the same formula, but have the different parameters (W_z, U_z) and (W_r, U_r) trained separately from each other.

After the recurrent operations, the final hidden state h_t at the end of a given input sequence is passed to the fully-connected prediction layer. It corresponds to the dialogue history features in this architecture and replaces the concatenated ones on $u_{t-w+1:t-2}$ by the CNN architecture. But the other local features $\vec{u}_t$ and $\vec{u}_{t-1}$ still come directly from the CNNs, which is common in both architectures.

Dynamic Memory Networks

Understanding human conversations with multiple topics contains the problem of tracing back to somewhere remote in dialogue history with regard to each subject in focus. For example, the topic transition at $t = 11$ in Figure 1 doesn't initiate a whole new subject, but just resumes the one that has already discussed by $t = 5$.

To incorporate the subject-specific long-term dependencies into dialogue topic tracking, we propose dynamic memory networks (Figure 4) to generate the dialogue history features as the recurrent architecture does in RCNN. The networks maintain a set of multiple memory slots each of which encodes the latent representation corresponding to an important subject on the domain of conversation. These memories are updated by recurrent operations going through a given sequence of utterance vectors. In this section, we present three different types of memory units in our proposed dynamic networks and compare them focusing on their gating mechanisms.

Memory with a single gate The first memory unit (Figure 5a) has a single gate function to update the hidden state of each memory slot, as follows:

$$z_i^j = \sigma\left(\vec{u}_i^T w^j + \vec{u}_i^T h_{i-1}^j\right), \quad (7)$$

$$\tilde{h}_i^j = \tanh\left(U h_{i-1}^j + V w^j + W \vec{u}_i\right), \quad (8)$$

$$h_i^j = \left(1 - z_i^j\right) \circ h_{i-1}^j + z_i^j \circ \tilde{h}_i^j, \quad (9)$$

where z_i^j is a gate function for the j-th memory slot at the i-th time step, w^j is a trainable key vector, $\tilde{h}_i^j$ is a candidate state, and the parameters U, V, and W are shared across all the slots. The gate function z_i^j is controlled by two terms $\vec{u}_i^T w^j$ and $\vec{u}_i^T h_{i-1}^j$ which correspond to the matchings from the input $\vec{u}_i$ to the key vector w^j and the previous slot state h_{i-1}^j, respectively.

This architecture is inspired by recurrent entity networks (Henaff et al. 2016) which achieved the state-of-the-art performances in question answering tasks. It also used a single gate function activated with the content-based matchings and maintains the long-term memory for each slot updated independently from the others in parallel.

Memory with update and reset gates While the memory network for question answering aims at referring to the answer of a given question from the past, our dynamic models for dialogue topic tracking learns the state representation at each moment of a given conversation. A distinct difference in solving the problems is that the outdated information needs to be filtered out from the topic tracking memories to keep the current state up to date. For example, the item suggested at $t = 3$ in Figure 1 has been immediately denied at $t = 4$, which would have a low probability of being discussed again in the same conversation. But the single gate architecture has a possibility that the outdated content remains somewhere in the memories, while the corresponding slots are not being updated with any new information.

To overcome this limitation, we propose another memory architecture (Figure 5b) with an additional reset gate defined as follows:

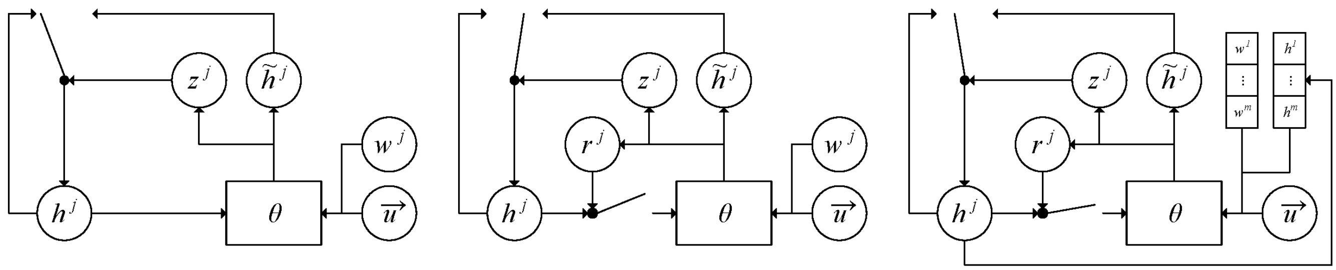
$$r_i^j = \sigma\left(\vec{u}_i^T W_r w^j + \vec{u}_i^T U_r h_{i-1}^j\right), \quad (10)$$

where W_r and U_r are the transform parameters to the same addressing terms $\vec{u}_i^T w^j$ and $\vec{u}_i^T h_{i-1}^j$ as in the update gate z_i^j. This reset gate is applied to the new candidate state as follows:

$$\tilde{h}_i^j = \tanh\left(U\left(r_i^j \circ h_{i-1}^j\right) + V w^j + W \vec{u}_i\right), \quad (11)$$

which is similar to GRU's reset gate usage.

Memory with cross-slot interactions The motivation of our final architecture comes from the hypothesis that any state update for a topic would affect also to the other topic states in dialogue context modelling. For example, in Figure 1, the attraction name is mentioned explicitly only in the earlier part of the conversation ($t = 5$) and another but related topic on 'transportation' is discussed in the following segment. We believe that a good tracker is supposed to keep the representation corresponding to this attraction until the conversation topic returns to the same subject again at $t = 11$. But it has to be phased out from $t = 13$, since the discussion about the particular attraction seems already finished according to the contexts in the new segment.



(a) with a single gate (b) with update and reset gates (c) with cross-slot interactions

Figure 5: Dynamic memory units for dialogue topic tracking

To incorporate this kind of correlations between different conversation subjects into the context representation with multiple memory slots, our model (Figure 5c) takes the cross-slot interactions by the following update and reset gate functions:

$$z_i^j = \sigma\left(\sum_k \left(\alpha_z^{kj} \vec{u}_i^T w^k + \beta_z^{kj} \vec{u}_i^T h_{i-1}^k\right)\right), \tag{12}$$

$$r_i^j = \sigma\left(\sum_k \left(\alpha_r^{kj} \vec{u}_i^T w^k + \beta_r^{kj} \vec{u}_i^T h_{i-1}^k\right)\right), \tag{13}$$

where α_z^{kj}, β_z^{kj}, α_r^{kj}, and β_r^{kj} are the coefficients to learn the correlations between the j-th and the k-th memory slots in updating and resetting the memories. Different from the distributed architectures in the previous sections, the key vectors $[w^1 \cdots w^m]$, the hidden states $[h^1 \cdots h^m]$ and all the other parameters are shared in the concurrent update across all the memory slots.

Evaluation

Data

To demonstrate the effectiveness of our proposed models, we conducted experiments on TourSG corpus which is the benchmark dataset for the fourth dialogue state tracking challenge (DSTC4) (Kim et al. 2016). It consists of 35 dialogue sessions each of which was collected from the conversations between a tour guide and a tourist about planning a trip to Singapore. Every dialogue session has been manually transcribed and annotated with the labels including the segmentation boundaries and the topic category for each segment into one of the following eight classes: 'attraction', 'transportation', 'food', 'accommodation', 'shopping', 'opening', 'closing', and 'other'. For our experiments, all these segment-level annotations were converted into the utterance-level B/I/O tags each of which belongs to one of 15 classes: $(\{\text{B-}, \text{I-}\} \times \{c : c \in C;\ \text{and } c \neq \text{'other'}\}) \cup \{\text{O}\}$, where C consists of all the eight topic categories. We used the same partition of the dataset (Table 1) as in the dialogue state tracking task in DSTC4.

Models

Based on the dataset, we built six neural network models categorized into three architecture families: CNNs, RCNNs, and dynamic memories.

Set	# sessions	# segments	# utterances
Train	14	2,104	12,759
Dev	6	700	4,812
Test	15	2,210	13,463
Total	35	5,014	31,034

Table 1: Statistics of TourSG dialogue corpus divided into three subsets for training, development, and test purposes.

The first baseline came from the CNN model which has achieved the best performances in our previous experiments (Kim, Banchs, and Li 2016) on the same dataset. In this model, the word embedding was initialized with the 300-dimensional word vectors pre-trained by word2vec (Mikolov et al. 2010) on 2.9M sentences from the 553k travel forum posts about Singapore. The convolutional layer learned 100 feature maps for each of three different filter sizes [3, 4, 5] by sliding them over the current, the previous, and the history utterances within the window size $w = 10$, which generated 900 feature values in total after the max-pooling operations. In addition, the speakers of the current and the previous utterances were introduced as extra features in the fully-connected layer.

Then, we compared two variants of the RCNN architecture using a vanilla RNN and a GRU as the unit in the recurrent layers. The hidden layer dimensions were 150 for the vanilla RNN and 50 for the GRU which were chosen by the grid search on the development set. The earlier parts of both models for the embedding, convolution, and max pooling operations have the same configurations and parameters with the CNN baseline.

Finally, three dynamic memory networks were trained based on the proposed gating mechanisms. The number of memory slots determined on the development set were $m = 5$ for the first two distributed models and $m = 10$ for the other with cross-slot interactions.

All the models were trained with Adam optimizer (Kingma and Ba 2014) by minimizing the categorical cross entropy loss on softmax. In the training phase, we used mini-batch size of 50 and applied dropout after the max pooling layer with the rate of 0.25 for regularization. We stopped training every model after 150 epochs where the performance of CNN baseline has been saturated.

Models	Sequential Labelling P	R	F	Segmentation P_k	WD
CNN (Kim, Banchs, and Li 2016)	0.6691	0.6861	0.6775	0.3799	0.4884
RCNN with vanilla RNNs	0.6825	0.6572	0.6696	0.3970	0.4634
RCNN with GRUs	0.6936	0.6826	0.6880	0.3888	0.4619
Memories with a single gate	0.6877	**0.7105**	0.6989	0.3782	0.4393
Memories with reset and update gates	0.6959	0.7035	0.6997	0.3781	0.4427
Memories with cross slot interactions	**0.7008**	0.7090	**0.7049**†	**0.3532**‡	**0.4223**‡

Table 2: Comparisons of the topic tracking performances with different models on TourSG dialogues. The higher precision (P), recall (R), and F-measure (F) scores the better results in sequential labelling, while the lower P_k and WindowDiff (WD) values the more accurate segmentations. The best score for each metric is highlighted in bold face with or without the indicator † and ‡ of its statistical significance to the second-best result at $p < 0.05$ and $p < 0.01$, respectively.

Metrics

The evaluations were performed on two major criteria. The first one is the sequential labelling performances evaluated with precision, recall, and F-measure of the predictions as they are compared to the gold standard annotations encoded also with the B/I/O tagging scheme.

The other metrics focus only on the segmentation capabilities of the topic tracking models on the binary label converted from every predicted or reference label y_t as follows:

$$y'_t = \begin{cases} 0 & \text{if } (y_t = \text{`I-}c\text{' and } y_{t-1} = \text{`B-}c\text{'}) \\ & \text{or } (y_t = \text{`I-}c\text{' and } y_{t-1} = \text{`I-}c\text{'}) \\ & \text{or } (y_t = \text{`O' and } y_{t-1} = \text{`O'}), \\ 1 & \text{otherwise.} \end{cases} \tag{14}$$

The boundary detection performances were evaluated with P_k (Beeferman, Berger, and Lafferty 1999) and WindowDiff (Pevzner and Hearst 2002), which are widely used metrics in segmentation that slide a fixed-size window through the reference and predicted sequences of segmentation boundaries and count the number of windows with mismatched boundaries for computing the scores. We set the window size $k = 3$ which is half of the average length of the reference segments in the test set.

Then, the statistical significance for each pair of models was computed using approximate randomization (Yeh 2000) for every metric.

Results

Table 2 compares the performances of the models trained on the combined data both from the training and development sets and evaluated on the test data set.

Comparing between two RCNN variants, the model with GRUs showed much better results than the other one using vanilla RNNs in all the metrics for both sequential labelling and segmentation. In addition, the GRUs in the recurrent layer also contributed to the significant improvement ($p < 0.01$) from the CNN baseline in F-measure, while the vanilla RNNs adversely affected to the sequential labelling performances. But the benefits by the RCNN architecture were not proven consistently enough for segmentation, since both RCNN models were evaluated as better in WindowDiff, but at the same time, worse in P_k than the CNN baseline, with statistical significances, $p < 0.001$ and $p < 0.01$, respectively.

Whereas, our proposed dynamic memory networks demonstrated the impacts to the overall topic tracking performances not only for sequential labelling, but also for segmentation. All the three memory architectures achieved better scores than the CNN and RCNN models in F-measure and both segmentation metrics. Especially, each of the improvements in F-measure by the proposed models was statistically significant ($p < 0.001$) from every baseline.

The first dynamic memory model with a single update gate produced the outcomes with a higher recall in sequential labelling than any other models in the evaluation also including the other two variants with dynamic memories. This enhanced coverage complemented its precision losses to get a higher score in F-measure, despite the even worse precision than the best RCNN model with GRU. The improvements in segmentation were also statistically significant at $p < 0.05$ except the one in P_k from the CNN baseline.

However, the additional reset gate introduced to the distributed architecture failed to make a distinctive contribution to the topic tracking performances. Although the second model with both the update and reset gates obtained slightly higher scores in F-measure and P_k than the single gate architecture, it caused loss in the segmentation performance measured by WindowDiff. But all these differences between two models were not statistically significant.

On the other hand, the results by our final model showed the effectiveness of the proposed architecture considering the cross-slot interactions for both updating and resetting memories. The model achieved the best performances against all the others in every metric except recall. Even for this exception, it had the second-best score in recall with no significant difference from the first one by the single-gated memories. All the improvements in F-measure by this model passed the statistical significance tests at $p = 0.05$ from the other dynamic memory networks and $p = 0.001$ from the CNN and RCNN baselines. The differences of the segmentation performances were also significant ($p < 0.001$) in both metrics without exception.

Figure 6 presents a heat map of the update gate z_i^j values in the cross-slot interaction model for each pair of memory slots and predicted labels. These slot-label correlations indicate that multiple memory slots are involved together in predicting a single label. And each label is associated with

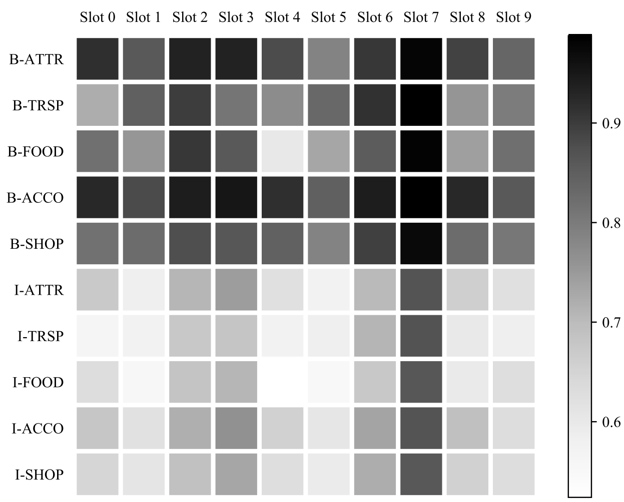


Figure 6: Mean values of the update gate z_i^j in the model with cross-slot interactions for each pair of the predicted label at i-th time step and the j-th memory slot.

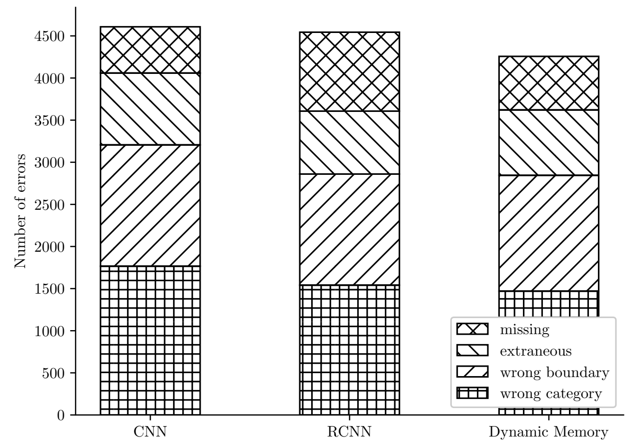


Figure 7: Distributions of the errors generated from the best model of each architecture: CNN baseline, RCNN with GRUs, and dynamic memories with cross slot interactions.

a particular set of slots which differs from other labels. As expected, the memory states are more dynamically updated at the beginning of each segment than the others inside.

Figure 7 shows the distributions of the errors generated by three models each of which reported the best results from its corresponding architecture family. Following the error analysis in (Kim, Banchs, and Li 2016), each erroneous prediction was categorized into the following four error types:

- Missing predictions: when the reference belongs to one of the labels other than ‘O’, but the model predicts it as ‘O’.
- Extraneous labelling: when the reference belongs to ‘O’, but the model predicts it as another label.
- Wrong categorizations: when the reference belongs to a category other than ‘O’, but the model predicts it as another wrong category.
- Wrong boundary detections: when the model outputs the correct category, but with a wrong prediction from ‘B’ to ‘I’ or from ‘I’ to ‘B’.

The distributions indicate that the reduced numbers of wrong categories were the decisive factor in performance improvements by the recurrent architectures in RCNN and dynamic memory networks compared to the CNN baseline. In addition, the difference in missing predictions shows that the dynamic memories have the better capabilities than RCNN in distinguishing between ‘O’ and the other positive labels.

Conclusions

This paper presented dynamic memory networks for dialogue topic tracking with three different gating mechanisms. The architectures were designed to learn the subject-specific long-term dependencies in dialogue sequences by updating multiple memory slots each of which corresponds to the state representation of a latent subject. Experimental results showed that the proposed approaches contributed to improve both the sequential labelling and segmentation performance with respect to the CNN and RCNN baselines.

The main hypothesis that the proposed dynamic memory networks are capable of representing better dialogue states of human conversations with multiple topics has been proven for the dialogue topic tracking task in this work. Our next step is exploring the models also for the multi-topic state tracking task (Kim et al. 2016) which aims at slot filling with more details about each subject in focus besides the topic category.

The other direction of our future work is to investigate the effective and efficient ways of incorporating external knowledge into the dialogue tracking models. Since many parts of human conversations have a high level of dependence on the domain knowledge that is not explicitly mentioned by the participants, we believe that what and how to leverage useful external resources will be a key to further advancement of dialogue topic and state tracking technologies.

References

Adams, P. H., and Martell, C. H. 2008. Topic detection and extraction in chat. In *Proceedings of the 2008 IEEE International Conference on Semantic Computing*, 581–588.

Beeferman, D.; Berger, A.; and Lafferty, J. 1999. Statistical models for text segmentation. *Machine learning* 34(1-3):177–210.

Bohus, D., and Rudnicky, A. 2003. Ravenclaw: dialog management using hierarchical task decomposition and an expectation agenda. In *Proceedings of the European Conference on Speech, Communication and Technology*, 597–600.

Celikyilmaz, A.; Hakkani-Tür, D.; and Tür, G. 2011. Approximate inference for domain detection in spoken language understanding. In *Proceedings of the 12th Annual*

Conference of the International Speech Communication Association (INTERSPEECH), 713–716.

Cho, K.; van Merrienboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR* abs/1409.1259.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research* 12:2493–2537.

Esteve, Y.; Bouallegue, M.; Lailler, C.; Morchid, M.; Dufour, R.; Linares, G.; Matrouf, D.; and De Mori, R. 2015. Integration of word and semantic features for theme identification in telephone conversations. In *Natural Language Dialog Systems and Intelligent Assistants*. Springer. 223–231.

Graves, A.; Wayne, G.; Reynolds, M.; Harley, T.; Danihelka, I.; Grabska-Barwińska, A.; Colmenarejo, S. G.; Grefenstette, E.; Ramalho, T.; Agapiou, J.; et al. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature* 538(7626):471.

Graves, A.; Wayne, G.; and Danihelka, I. 2014. Neural turing machines. *CoRR* abs/1410.5401.

Henaff, M.; Weston, J.; Szlam, A.; Bordes, A.; and LeCun, Y. 2016. Tracking the world state with recurrent entity networks. *CoRR* abs/1612.03969.

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

Ikeda, S.; Komatani, K.; Ogata, T.; Okuno, H. G.; and Okuno, H. G. 2008. Extensibility verification of robust domain selection against out-of-grammar utterances in multi-domain spoken dialogue system. In *Proceedings of the 9th INTERSPEECH*, 487–490.

Kalchbrenner, N.; Grefenstette, E.; and Blunsom, P. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 655–665.

Kim, S.; Banchs, R. E.; and Li, H. 2014. A composite kernel approach for dialog topic tracking with structured domain knowledge from wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 19–23.

Kim, S.; Banchs, R.; and Li, H. 2016. Exploring convolutional and recurrent neural networks in sequential labelling for dialogue topic tracking. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 963–973.

Kim, S.; D'Haro, L. F.; Banchs, R. E.; Williams, J. D.; and Henderson, M. 2016. The fourth dialog state tracking challenge. In *Proceedings of the 7th International Workshop on Spoken Dialogue Systems (IWSDS)*.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Lagus, K., and Kuusisto, J. 2002. Topic identification in natural language dialogues using neural networks. In *Proceedings of the 3rd SIGdial workshop on Discourse and dialogue*, 95–102.

Lee, J. Y., and Dernoncourt, F. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 515–520.

Lee, C.; Jung, S.; and Lee, G. G. 2008. Robust dialog management with n-best hypotheses using dialog examples and agenda. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 630–637.

Lin, B.; Wang, H.; and Lee, L. 1999. A distributed architecture for cooperative spoken dialogue agents with coherent dialogue state and history. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.

Mikolov, T.; Karafiát, M.; Burget, L.; Cernockỳ, J.; and Khudanpur, S. 2010. Recurrent neural network based language model. In *INTERSPEECH*, volume 2, 3.

Morchid, M.; Dufour, R.; Bouallegue, M.; Linares, G.; and De Mori, R. 2014a. Theme identification in human-human conversations with features from specific speaker type hidden spaces. In *INTERSPEECH*, 248–252.

Morchid, M.; Dufour, R.; Bousquet, P.; Bouallegue, M.; Linares, G.; and De Mori, R. 2014b. Improving dialogue classification using a topic space representation and a gaussian classifier based on the decision rule. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 126–130. IEEE.

Nakata, T.; Ando, S.; and Okumura, A. 2002. Topic detection based on dialogue history. In *Proceedings of the 19th international conference on Computational linguistics (COLING)*, 1–7.

Pevzner, L., and Hearst, M. A. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics* 28(1):19–36.

Ramshaw, L. A., and Marcus, M. P. 1995. Text chunking using transformation-based learning. In *Proceedings of the 3rd Workshop on Very Large Corpus*, 88–94.

Roy, S., and Subramaniam, L. V. 2006. Automatic generation of domain models for call centers from noisy transcriptions. In *Proceedings of COLING/ACL*, 737–744.

Shen, Y.; He, X.; Gao, J.; Deng, L.; and Mesnil, G. 2014. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web (WWW)*, 373–374. International World Wide Web Conferences Steering Committee.

Yeh, A. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, 947–953.