Exploring Convolutional and Recurrent Neural Networks in Sequential Labelling for Dialogue Topic Tracking

Seokhwan Kim, Rafael E. Banchs, Haizhou Li
Human Language Technology Department
Institute for Infocomm Research
Singapore 138632
{kims,rembanchs,hli}@i2r.a-star.edu.sg

Abstract

Dialogue topic tracking is a sequential labelling problem of recognizing the topic state at each time step in given dialogue sequences. This paper presents various artificial neural network models for dialogue topic tracking, including convolutional neural networks to account for semantics at each individual utterance, and recurrent neural networks to account for conversational contexts along multiple turns in the dialogue history. The experimental results demonstrate that our proposed models can significantly improve the tracking performances in human-human conversations.

1 Introduction

A human conversation often involves a series of multiple topics contextually related to each other. In this scenario, every participant in the conversation is required to understand the on-going topic discussed at each moment, detect any topic shift made by others, and make a decision to self-initiate a new topic. These human capabilities for handling topics are also expected from dialogue systems to achieve natural and human-like conversations.

Many studies have been conducted on multi-domain or multi-task dialogue systems by means of sentence-level topic identification as a subtask of natural language understanding (Lin et al., 1999; Nakata et al., 2002; Lagus and Kuusisto, 2002; Adams and Martell, 2008; Ikeda et al., 2008; Celikyilmaz et al., 2011). In these approaches, a given user input at each turn is categorized into topic classes, each of which triggers the corresponding sub-system specializing in the particular topic. Despite many previous efforts, the sentence categorization methods still have the following limitations. Firstly, the effectiveness of the approaches is limited only in user-initiative conversations, because the categorization is performed mainly based on the user's input mentioned at a given turn. Secondly, no correlation between different topics is considered neither in the topic decision process nor in each topic-specific sub-system operated independently from the others. Lastly, the conversational coherence in a given dialogue history sequence has limited effects on determining the current topic.

Another direction for multi-topic dialogue systems has been towards utilizing human knowledge represented in domain models (Roy and Subramaniam, 2006) and agendas (Bohus and Rudnicky, 2003; Lee et al., 2008). The knowledge-based approaches make the system capable of having more control of dialogue flows including topic sequences. This aspect contributes to better decisions of topics in system-initiative cases, but it can adversely affect the flexibility to deal with unexpected inputs against the system's suggestions. Moreover, the high cost of building the required resources is another problem that these methods face from a practical point of view.

Recently, some researchers (Morchid et al., 2014a; Morchid et al., 2014b; Esteve et al., 2015) have worked on topic identification for analyzing human-human dialogues. Although they don't aim at building components in dialogue systems directly, the human behaviours learned from the conversations can suggest directions for further advancement of conversational agents. However, the problem defined in the studies is under the assumption that every dialogue session is assigned with just a single theme category, which means any topic shift occurred in a session is left out of consideration in the analyses.

On the other hand, we previously addressed the problem of detecting multiple topic transitions

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 963–973,
Berlin, Germany, August 7-12, 2016.

in mixed-initiative human-human conversations, which is called dialogue topic tracking (Kim et al., 2014a; Kim et al., 2014b). In these studies, the tracking task is formulated as a classification problem for each utterance-level, similar to the sentence categorization approaches. But the target of the classification is not just an individual topic category to which each input sentence belongs, but the decision whether a topic transition occurs at a given turn as well as what the most probable topic category will follow after the transition.

This paper presents our work also on dialogue topic tracking mainly focusing on the following issues. Firstly, in addition to transitions between dialogue segments from different topics, transitions between segments belonging to the same topic are also detected. This focuses the task more on detailed aspects of topic handling that are relevant to other subtasks such as natural language understanding and dialogue state tracking, rather than the conventional tracking of changes in topic categories only. Another contribution of this work is that we introduce a way to use convolutional neural networks in topic tracking to improve the classification performances with the learned convolutional features. In addition, we also propose the architectures based on recurrent neural networks to incorporate the temporal coherence that has not played an important role in previous approaches.

The remainder of this paper is structured as follows. We present a problem definition of dialog topic tracking in Section 2. We describe our proposed approaches to this task using convolutional and recurrent neural networks in Section 3. We report the evaluation result of the methods in Section 4 and conclude this paper in Section 5.

2 Dialogue Topic Tracking

Dialogue topic tracking is defined as a multi-class classification problem to categorize the topic state at each time step into the labels encoded in BIO tagging scheme (Ramshaw and Marcus, 1995) as follows:

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

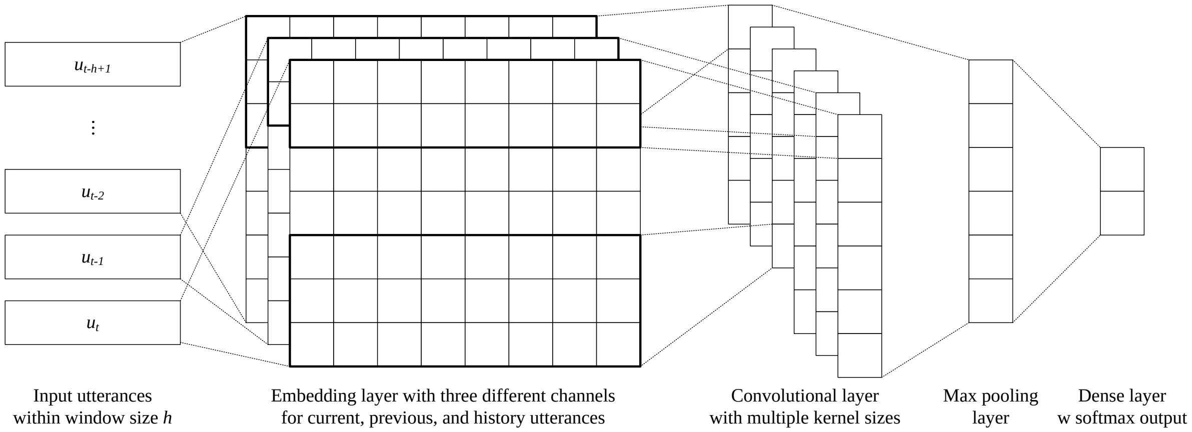
where u_t is the t-th utterance in a given dialogue session and C is a closed set of topic categories.

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	Or, you can use the shuttle bus service from the hotels in Orchard, which is free of charge.	B-TRSP
12	Tourist	Great! That would be definitely better.	I-TRSP
13	Guide	After visiting the park, you can enjoy some seafoods at the riverside on the way back.	B-FOOD
14	Tourist	What food do you have any recommendations to try there?	I-FOOD
15	Guide	If you like spicy foods, you must try chilli crab which is one of our favourite dishes.	I-FOOD
16	Tourist	Great! I'll try that.	I-FOOD

Figure 1: Examples of dialogue topic tracking on a tour guide dialogue labelled with BIO tags. ATTR, TRSP and FOOD denotes the topic categories of attraction, transportation, and food, respectively.

Figure 1 shows an example of topic tracking on a dialogue fragment between a tour guide and a tourist. Since each tag starting with 'B' should occur at the beginning of a new segment after a topic transition from its previous one, the label sequence indicates that this conversation is divided into six segments at $t = \{2, 4, 6, 11, 13\}$. The initiativity of each segment can be also found from who the speaker of the first utterance of the segment is. In this example, three of the cases are initiated by the tourist at $t = \{2, 4, 6\}$, but the others are leaded by the tour guide, which means it is a mixed-initiative type of conversation.

Different from the former studies (Kim et al., 2014a; Kim et al., 2014b) that were only focused on detecting transitions between different topic categories, this work subdivides each dialogue sequence which belongs to a single topic category, but discusses more than one subject that can be more specifically differentiated from each other. The above example also has two cases of transitions with no change of topic categories at $t = \{4, 11\}$: the first one is due to the tourist's request for an alternative attraction from the recommendation in the previous segment, and the other transition is triggered by the tour guide to suggest another option of transportation which is also available for the route discussed previously.



u_{t-h+1}
⋮
u_{t-2}
u_{t-1}
u_t
Input utterances within window size h
Embedding layer with three different channels for current, previous, and history utterances
Convolutional layer with multiple kernel sizes
Max pooling layer
Dense layer w softmax output

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

3 Models

The classifier f can be built with supervised machine learning techniques, when a set of example dialogues manually annotated with gold standard labels are available as a training set. The earlier studies (Kim et al., 2014a; Kim et al., 2014b) also proposed supervised classification approaches particularly focusing on kernel methods to incorporate domain knowledge obtained from external resources into the linear vector space models based on bag-of-words features extracted from the training dialogues.

This work, on the other hand, aims at improving the classification capabilities only with the internal contents in given dialogues rather than making better uses of external knowledge. To overcome the limitations of the simple vector space models used in the previous work, we propose models based on convolutional and recurrent neural network architectures. These models are presented in the remainder of this section.

3.1 Convolutional Neural Networks

A convolutional neural network (CNN) automatically learns the filters in its convolutional layers which are applied to extract local features from inputs. Then, these lower-level features are combined into higher-level representations following a given network architecture. These aspects of CNNs make themselves a good fit to solve the problems which are invariant to the location where each feature is extracted on its input space and also depend on the compositional relationships between local and global features, which is the reason why CNNs have succeed in computer vision (LeCun et al., 1998). As implied by the successes of bag-of-words or bag-of-ngrams considering the existence of each linguistic unit independently and the important roles of compositional structures in linguistics, CNN models have recently achieved significant improvements also in some natural language processing tasks (Collobert et al., 2011; Shen et al., 2014; Yih et al., 2014; Kalchbrenner et al., 2014a; Kim, 2014).

The model for dialogue topic tracking (Figure 2) is basically based on the CNN architecture proposed by Collobert et al. (2011) and Kim (2014) for sentence classification tasks. In the architecture, a sentence of length n is represented as a matrix with the size of $n \times k$ concatenated with n rows each of which is the k-dimensional word vector $\vec{x}_i \in \mathbb{R}^k$ representing the i-th word in the sentence. This embedding layer can be learned from scratch with random initialization or fine-tuned from pre-trained word vectors (Mikolov et al., 2013) with back propagation during training the network.

Unlike other sentence classification tasks, dialogue topic tracking should consider not only a single sentence given at each time step, but also the other utterances previously mentioned. To incorporate the dependencies to the dialogue history into the topic tracking model, the input at the time step t is composed of three different channels each of which represents the current utterance u_t, the previous utterance u_{t-1}, and the other utterances $u_{t-h+1:t-2}$ within h time steps, respectively, where u_t is the t-th utterance in a session, $u_{i:j}$ is the concatenation of the utterances occurred from the i-th to the j-th time steps in the history, and h is the size of history window. The height of the $n \times k$ matrices of the first two channels for the current and previous utterances is fixed to the length of the longest utterance in the whole training dataset, and then all the remaining rows after the end of each utterance are zero-padded to make all inputs same size. Since the other channel is made up by concatenating the utterances from the $(t - h + 1)$-th to the $(t - 2)$-th time steps, it has a matrix with the dimension of $((h - 2) \cdot n) \times k$ where all the gaps between contiguous utterances in the matrix are filled with zero.

In the convolutional layer, each filter $\mathcal{F} \in \mathbb{R}^{km}$ which has the same width k as the input matrix and a given window size m as its height slides over from the first row to the $(n - m + 1)$-th row of the input matrix. At the i-th position, the filter is applied to generate a feature $c_i = g\left(\mathcal{F} \cdot \vec{x}_{i:i+m-1} + b\right)$, where $\vec{x}_{i:j}$ is the subregion from the i-th row to the j-th row in the input, $b \in \mathbb{R}$ is a bias term, and g is a non-linear activation function such as rectified linear units. This series of convolution operations 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 value $c' = \max(\vec{c})$ is selected from each feature map considered as the most important feature for the particular filter in the max-pooilng layer.

Every filter is shared across all the three different channels, but both the convolution and max-pooling operations are performed individually for each channel. Thus, the total number of feature values generated in the pooling layer is three times the number of filters. Finally, these values are forwarded to the fully-connected layer with softmax which generates the probabilistic distribution over the topic labels for a given input.

3.2 Recurrent Neural Networks

Dialogue topic tracking is conceptually performed on a sequence of interactions exchanged by the participants in a given session from its beginning to each turn. Thus, the contents discussed previously in the dialogue history are likely to have an important influence on tracking the current topic at a given turn, which is a fundamental difference from other text categorization problems that consider each input independently from all others.

To make use of the sequential dependencies in dialogue topic tracking, we propose the models based on recurrent neural networks (RNN) which learn the temporal dynamics by recurrent computations applied to every time step in a given input sequence. In a traditional RNN, hidden states connecting between input sequences and output labels are repeatedly updated with the operation $\vec{s_t} = g(Ux_t + Ws_{t-1})$, where x_t is the t-th element in a given input sequence, $\vec{s_t} \in \mathbb{R}^{|s|}$ is the hidden state at t with $|s|$ hidden units, and g is a non-linear activation function. The parameters U and W are shared all the time steps.

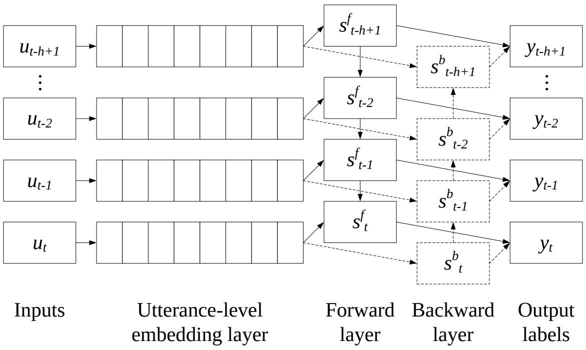


Figure 3: Recurrent neural network architecture for dialogue topic tracking. The backward layer with the dotted lines is enabled only with its bidirectional extension.

RNNs have been successfully applied to several natural language processing tasks including language modeling (Mikolov et al., 2010), text generation (Sutskever et al., 2011), and machine translation (Auli et al., 2013; Liu et al., 2014), all of which focus on dealing with variable length word sequences. On the other hand, an input sequence to be handled in dialogue topic tracking is composed of utterance-level units instead of words.

In our model (Figure 3), each utterance is represented by the k-dimensional vector $\vec{u}_t \in \mathbb{R}^k$ assigned with pre-trained sentence-level embeddings (Le and Mikolov, 2014). And then, a sequence of the utterance vectors within h time steps are connected in the recurrent layers. The default sequence of applying the recurrent operation is the ascending order from the former to the recent utterances, which is performed in the forward layer. But the opposite direction can be also considered in the backward layer which is stacked on top of the forward layer to build a bidirectional RNN (Schuster and Paliwal, 1997) which outputs the concatenation of both forward and backward states as an outcome of the recurrent operations. Then, these hidden states from the recurrent layers are passed to the fully-connected softmax layer to generate the output distributions for every time step in the sequence.

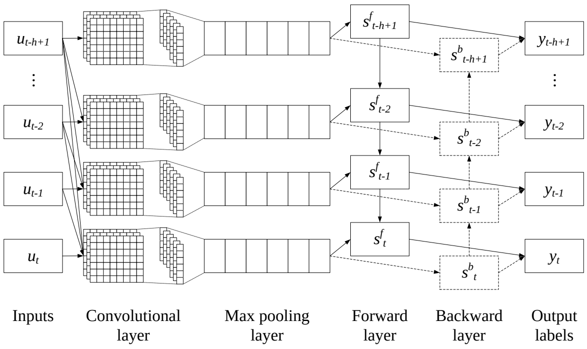


Figure 4: Recurrent convolutional network architecture for dialogue topic tracking. The backward layer is only for the bi-directional mode.

The output from the model at a given time step t is a label sequence $[y_{t-h+1}, \cdots, y_t]$ for the recent h utterances. Since the labels for the earlier utterances should have been already decided at the corresponding turns, only y_t is taken as the final outcome for the current time step. The hypothesis to be examined with this model is whether the other $h-1$ predictions that are not directly reflected to the results could help to improve the tracking performances by being considered together in the process of determining the current topic status.

3.3 Recurrent Convolutional Networks

The last approach proposed in this work aims at combining the two models described in the previous sections. In this model (Figure 4), each feature vector generated through the embedding, convolutional, and max pooling layers in the CNN network (Section 3.1) is connected to the recurrent layers in the RNN model (Section 3.2). This combination is expected to play a significant role in overcoming the limitations of the sentence-level embedding considered as a feature representation in the RNN model. While the previous approach depends only on a pre-trained and non-tunable embedding model, all the parameters in the combined network can be fine-tuned with back propagation by considering the convolutional features extracted at each time step and also the temporal dependencies occurred through multiple time steps in given dialogue sequences.

In computer vision, this kind of models connecting RNNs on top of CNNs is called recurrent convolutional neural networks (RCNN), which have been mostly used for exploring the dependencies between local convolutional features within a single image (Pinheiro and Collobert, 2014; Liang and Hu, 2015). Recently, they are also applied in video processing (Donahue et al., 2015) where visual features are extracted from the image at each frame using CNNs and the temporal aspects are learned with RNNs from the frame sequence of an input video. Our proposed model for dialogue topic tracking was originally motivated by this success of RCNNs particularly in video recognition considering that video and dialogue are analogous from the structural point of view. Each instance of a video and a dialogue consists of a temporal sequence of static units.

4 Evaluation

4.1 Data

To demonstrate the effectiveness of our proposed models, we performed experiments on TourSG corpus released for the fourth dialogue state tracking challenge (DSTC4) (Kim et al., 2016). The dataset consists of 35 dialogue sessions collected from human-human conversations about tourism in Singapore between tour guides and tourists. All the dialogues have been manually transcribed and annotated with the labels for the challenge tasks. For the multi-topic dialogue state tracking which is the main task of the challenge, each dialogue session is divided into sub-dialogues and each segment is assigned with its topic category. Since the task particularly focuses on filling out the topic-specific frame structure with the detailed information representing the dialogue states of a given segment, it has been performed under the assumption that the manual annotations for both segmentations and topic categories are provided as parts of every input. But, in this work for dialogue topic tracking, these labels are considered as the targets to be generated automatically by the models.

Every segment in the dataset belongs to one of eight topic categories. Following the nature of the tourism domain, the 'attraction' category accounts for the highest portion at 40.12% of the segments, which is followed by 'transportation', 'food', 'accommodation', 'shopping', 'closing' and 'opening' in order according to decreasing frequencies. The other 10.53% considered as beyond the scope of the task are annotated with 'other'.

Figure 5 shows the distributions of the segments by not only the topic categories, but also the transition types from two different points of views: the first one is which speaker initiates each segment, and the other is whether the segmentation causes

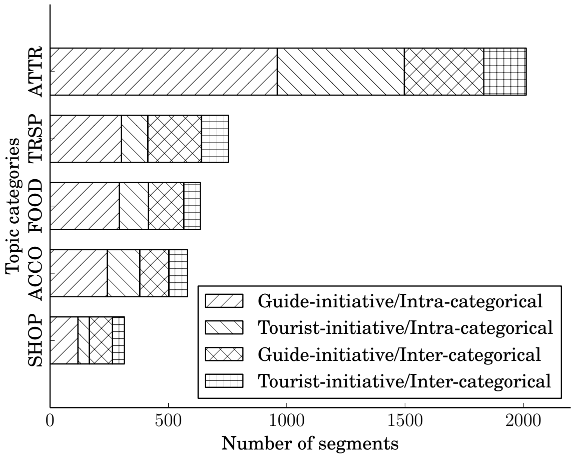


Figure 5: Distributions of the segments in TourSG corpus by topic categories and transition types. ATTR, TRSP, FOOD, ACCO and SHOP denotes the topic categories of attraction, transportation, food, accommodation, and shopping, respectively.

Table 1: Statistics of TourSG corpus. The whole dataset is divided into three subsets for training, development, and test purposes.

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

a topic category shift or not. The most frequent type found in the dataset is guide-initiative and intra-categorical transitions. 63.86% and 61.31% of the total segments are initiated by guides and segmented keeping topic categories, respectively.

For our experiments, all these segment-level annotations were converted into utterance-level BIO 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. The partition of the dataset (Table 1) have been kept the same as the one used for the state tracking task in DSTC4.

4.2 Models

Based on the dataset, we built 16 different models classified into the following five model families.

4.2.1 Baseline 1: Support Vector Machines

The first baseline uses support vector machine (SVM) (Cortes and Vapnik, 1995) models trained with the following features:

- BoN_t: bag of uni/bi/tri-grams in u_t weighted by tf-idf which is the product of term frequency in u_t and inverse document frequency across all the training utterances.
- BoN_{t-1}: bag of n-grams computed in the same way as BoN_t for the previous utterance.
- $\text{BoN}_{\text{history}} = \sum_{j=0}^{h} \left(\lambda^j \cdot \text{BoN}_j\right)$: weighted sum of n-gram vectors in the recent $h = 10$ utterances with a decay factor $\lambda = 0.9$.
- SPK_t, SPK_{t-1}: speakers of the current and the previous utterances.
- $\text{SPK}_{\{t-1,t\}}$: bi-gram of SPK_t and SPK_{t-1}.

Another variation replaces the bag of n-grams with the utterance-level neural embeddings inferred by the pre-trained 300 dimensional doc2vec (Le and Mikolov, 2014) model on 2.9M sentences with 37M words in 553k Singapore-related posts collected from travel forums. Then, the third model takes the concatenation of both bag of n-grams and doc2vec features.

All three baselines were implemented based on the one-against-all approach with the same number of binary classifiers as the total number of classes for multi-label classification. SVM*light* (Joachims, 1999) was used for building each binary classfier with the linear kernel.

4.2.2 Baseline 2: Conditional Random Fields

To incorporate the temporal aspects also into the linear models, conditional random fields (CRFs) (Lafferty et al., 2001) which have been successfully applied for other sequential labelling problems were used for the second set of baselines. Similar to our proposed RNN architecture (Section 3.2), the recent utterances occurred within the window size of $h = 10$ composed the first-order linear-chain CRFs. Three CRF models were built using CRFsuite (Okazaki, 2007) with the same feature sets as in the SVM models.

4.2.3 CNN-based models

For the CNN architecture (Section 3.1), we compared two different models: the first one learned the word embeddings from scratch with random parameters, while the other was initialized with word2vec (Mikolov et al., 2013) trained on the same dataset for the doc2vec model in Section 4.2.1. Both approaches generated a dense vector with a dimension of $k = 300$ for each word in utterances. Then, the embedded vectors were concatenated into three matrices representing the current, previous, and history utterances, respectively. While the first two channels for a single utterance, u_t or u_{t-1}, had a size of 65×300 according to the maximum number of words $n = 65$ in the training utterances, the number of rows in the other matrix was 520 which is eight times as large as the others to represent the history utterances from u_{t-9} to u_{t-2} where $h = 10$.

In the convolutional layer, 100 feature maps were learned for each of three different filter sizes $m = \{3, 4, 5\}$ by sliding them over the utterances, which produced 900 feature values in total after the max-pooling operations for all three channels. In addition to these learned features, SPK_t and SPK_{t-1} values introduced in Section 4.2.1 were appended to each feature vector to take the speaker information into account as in the baselines. Before the fully-connected layer, dropout was performed with the rate of 0.25 for regularization. And then, training was done with stochastic gradient descent (SGD) by minimizing categorical cross entropy loss on the training set.

All the neural network-based models in this work were implemented using Theano (Bergstra et al., 2010) with the parameters obtained from the grid search on the development set.

4.2.4 RNN and RCNN-based models

Each proposed recurrent network (Section 3.2 and 3.3) was implemented with four variations categorized by whether the backward layer is included in each model or not and also which architecture is used in the recurrent layers between traditional RNNs and long short-term memories (LSTMs) (Hochreiter and Schmidhuber, 1997). The RCNN models based on LSTMs are particularly called long-term recurrent convolutional networks (LRCN) (Donahue et al., 2015). All the RCNN-based models were initialized with the pre-trained word2vec model in the training phase.

The dimension of the hidden layers of the recurrent cells was chosen to be $|s| = 500$ based on the development set. And the other settings including the parameters, the training algorithm, and the loss fuction were the same as in Section 4.2.3.

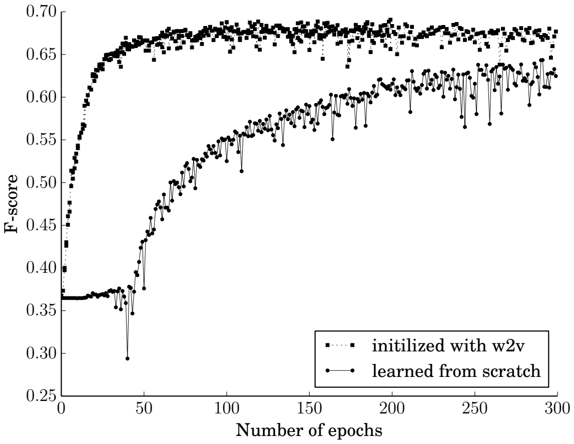


Figure 6: Comparisons of the topic tracking performances of the CNN models with different word embedding approaches according to the number of epochs for training in the development phase.

4.3 Results

Table 2 compares the performances of the models trained on the combination of the training and development sets and evaluated on the test set. The parameters for each model were decided in the development phase which built the models under various different settings only on the training set and validated them with the development set. The evaluations were performed with precision, recall, and F-measure to the manual annotations under three different schedules at tourist turns, guide turns, and all the turns. Then, the statistical significance for every pair was computed using approximate randomization (Yeh, 2000).

Comparing between two baseline families, the sequential extensions with the CRF models contributed to significant improvements ($p < 0.05$) from the SVM models in all the schedules. But in both SVM and CRF models, doc2vec features failed to achieve comparable performances to the simplest bag-of-ngrams features. Even the improvements by combining them to the word features were not statistically significant.

While these sentence-level embeddings trained in the unsupervised manner exposed the limitations in dialogue topic tracking performances, our proposed CNN-based models outperformed all these baselines. Especially, the CNN initialized with the pre-trained word2vec model achieved higher performances by 8.38%, 6.41%, and 7.21% in F-measure under each schedule, respectively, than the best baseline results.

	Schedule: Tourist Turns			Schedule: Guide Turns			Schedule: All		
Models	P	R	F	P	R	F	P	R	F
SVM (BoN+SPK)	61.60	62.18	61.89	58.65	58.42	58.53	59.85	59.94	59.90
SVM (D2V+SPK)	45.05	51.32	47.98	47.78	52.98	50.24	46.66	52.31	49.32
SVM (BoN+SPK+D2V)	61.60	62.18	61.89	58.74	58.53	58.63	59.91	60.01	59.96
CRF (BoN+SPK)	61.18	62.72	61.94	59.27	59.78	59.52	60.05	60.97	60.51
CRF (D2V+SPK)	61.53	49.42	54.81	61.94	49.68	55.13	61.77	49.57	55.00
CRF (BoN+SPK+D2V)	61.22	62.76	61.98	59.30	59.81	59.55	60.08	61.00	60.54
CNN (from scratch)	64.74	63.46	64.10	63.29	62.48	62.88	63.88	62.87	63.37
CNN (with W2V)	69.26	71.49	70.36	65.29	66.65	65.96	66.91	68.61	67.75
Uni-directional RNN	49.46	54.34	51.79	49.54	53.36	51.38	49.51	53.75	51.55
Bi-directional RNN	48.54	49.96	49.24	48.86	49.72	49.29	48.73	49.82	49.27
Uni-directional LSTM	49.52	50.81	50.15	49.41	49.85	49.63	49.45	50.23	49.84
Bi-directional LSTM	48.39	49.05	48.72	48.44	48.58	48.51	48.42	48.77	48.59
Uni-directional RCNN	69.49	71.59	70.52	65.43	66.68	66.05	67.08	68.67	67.86
Bi-directional RCNN	69.81	72.50	71.13	65.49	67.28	66.37	67.25	69.39	68.30
Uni-directional LRCN	69.37	71.45	70.40	**66.22**	67.41	66.81	67.50	69.04	68.26
Bi-directional LRCN	**69.85**	**72.56**	**71.18**	66.04	**67.62**	**66.82**	**67.60**	**69.62**	**68.59**

Table 2: Comparisons of the topic tracking performances with different models. D2V and W2V denote the vectors from doc2vec and word2vec, respectively.

Figure 6 presents the differences between two CNN models observed in the development phase. As the number of epochs increases, the performances of both models also increase up to certain points of saturation. But the model with random initialization required much longer time to be ready to gain scores in earlier iterations and its saturated performance was also lower than the other one learned on top of word2vec.

In contrast to the success of the CNN models, the proposed RNN architectures were not able to produce quality results, which was also caused by the limitations of doc2vec representations as already shown in the baseline results. Although some RNN models showed little performance gains over the SVM baselines only with doc2vec features, they were even worse than the CRF model with the same features.

On the other hand, the RCNN models connecting the results of CNNs to the RNNs contributed to performance improvements not only from the baselines, but also from the CNN models. While the uni-directional RNN was preferred in the RNN models only with doc2vec, the bi-directional LSTM showed better results in the RCNN architectures. As a result, the bi-directional LRCN model achieved the best performances against all the others, which were statistically significant ($p < 0.01$) compared to the second best results with bi-directional RCNN.

Table 3 shows the segmentation performances evaluated by considering only the beginning of each segment predicted by the best model of each architecture family. The proposed CNN and LRCN models demonstrated better capabilities of detecting topic transitions in both intra-categorical and inter-categorical conditions than the baselines. While the CNN model tended to have a higher coverage in segmentation than the others, the LRCN model produced more precise decisions to recognize the boundaries on the strength of the consideration of conversational coherences in dialogue history sequences.

However, the segmentation performances even with the best models were still very limited especially for inter-categorical transitions. And most of the models in the experiment had better performances in tourist turns than guide turns, as shown in Table 2. Considering the general characteristics of the target domain conversations that guide-driven and inter-categorical transitions are more likely to be dependent on human background knowledge than tourist-driven and intra-categorical cases, respectively, the current limitations are expected to be tackled by leveraging external resources into the models in future.

Finally, the generated errors from the models were categorized into the following error types:

	Intra-categorical			Inter-categorical			All		
Models	P	R	F	P	R	F	P	R	F
SVM (BoN+SPK+D2V)	40.22	30.19	34.49	8.68	28.14	13.26	18.65	29.51	22.85
CRF (BoN+SPK+D2V)	36.42	25.92	30.28	11.57	24.40	15.70	21.58	25.41	23.34
CNN (with W2V)	41.25	**41.50**	**41.37**	17.02	**40.87**	24.03	28.06	**41.29**	33.41
Bi-directional LRCN	**44.82**	38.28	41.29	**17.87**	40.72	**24.84**	**29.41**	39.09	**33.57**

Table 3: Comparisons of the segmentation performances with different models.

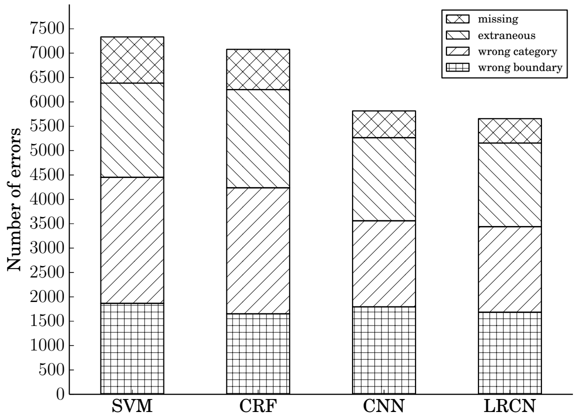


Figure 7: Distributions of errors generated from the best model of each architecture.

- 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 error distributions in Figure 7 indicate that the significantly decreased numbers of wrong categories were the decisive factor in performance improvements by our proposed approaches from the baselines. Besides, the enhanced capabilities of the models in distinguishing between ‘O’ and other labels were demonstrated by the reduced numbers of missing and extraneous predictions. The sequential architectures in CRF and LRCN models also showed its effectiveness especially in boundary detection, as expected.

5 Conclusions

This paper presented various neural network architectures for dialogue topic tracking. Convolutional neural networks were proposed to capture the semantic aspects of utterances given at each moment, while recurrent neural networks were intended to incorporate temporal aspects in dialogue histories into tracking models. Experimental results showed that the proposed approaches helped to improve the topic tracking performance with respect to the linear baseline models.

Furthering this work, there would be still much room for improvement in future. Firstly, the architectures based on a single convolutional layer and a single bi-directional recurrent layer in the proposed models can be extended by adding more layers as well as utilizing more advanced components including hierarchical CNNs (Kalchbrenner et al., 2014b) to deal with utterance compositionalities or attention mechanisms (Denil et al., 2012) to focus on more important segments in dialogue sequences.

Secondly, the use of external knowledge could be a key to success in dialogue topic tracking, as proved in the previous studies. However, this work only takes internal dialogue information into account for making decisions. If we develop a good way of leveraging other useful resources into the neural network architectures, better performance can be expected especially for guide-driven and inter-categorical topic transitions that are considered to be more dependent on background knowledge of the speakers.

The other direction of our future work is to investigate joint models for tracking dialogue topics and states simultaneously. Although the previous multi-topic state tracking task has assumed that the topics should be given as inputs to state trackers, we expect that a joint approach can contribute to both problems by dealing with the bi-directional relationships between them.

References

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

M. Auli, M. Galley, C. Quirk, and G. Zweig. 2013. Joint language and translation modeling with recurrent neural networks. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1044–1054.

J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. 2010. Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*, pages 1–7.

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

A. Celikyilmaz, D. Hakkani-Tür, and G. Tür. 2011. Approximate inference for domain detection in spoken language understanding. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 713–716.

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

C. Cortes and V. Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

M. Denil, L. Bazzani, H. Larochelle, and N. de Freitas. 2012. Learning where to attend with deep architectures for image tracking. *Neural computation*, 24(8):2151–2184.

J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634.

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

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

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

T. Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 169–184. MIT Press, Cambridge, MA.

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

N. Kalchbrenner, E. Grefenstette, and P. Blunsom. 2014b. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 655–665.

S. Kim, R. E. Banchs, and H. Li. 2014a. 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)*, pages 19–23.

S. Kim, R. E. Banchs, and H. Li. 2014b. Wikipedia-based kernels for dialogue topic tracking. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 131–135.

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

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

J. Lafferty, A. McCallum, and F.C.N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pages 282–289.

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

Q. Le and T. Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

C. Lee, S. Jung, and G. G. Lee. 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, pages 630–637.

M. Liang and X. Hu. 2015. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3367–3375.

B. Lin, H. Wang, and L. Lee. 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)*.

S. Liu, N. Yang, M. Li, and M. Zhou. 2014. A recursive recurrent neural network for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1491–1500.

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

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems (NIPS)*, pages 3111–3119.

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

M. Morchid, R. Dufour, P.M. Bousquet, M. Bouallegue, G. Linares, and R. De Mori. 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*, pages 126–130. IEEE.

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

N. Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs).

P. Pinheiro and R. Collobert. 2014. Recurrent convolutional neural networks for scene labeling. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 82–90.

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

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

M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45(11):2673–2681.

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

I. Sutskever, J. Martens, and G. E. Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024.

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

W. Yih, X. He, and C. Meek. 2014. Semantic parsing for single-relation question answering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 643–648.