

Improved Dynamic Memory Network for Dialogue Act Classification with Adversarial Training

Yao Wan^{*†}, Wenqiang Yan^{*†}, Jianwei Gao[†], Zhou Zhao[†], Jian Wu[†], Philip S. Yu^{†‡}

[†]College of Computer Science and Technology, Zhejiang University, Hangzhou, China

[‡]Department of Computer Science, University of Illinois at Chicago, Illinois, USA

[‡]Institute for Data Science, Tsinghua University, Beijing, China

{wanyao,gjwei,zhaozhou,wujian2000}@zju.edu.cn, yxlink2017@gmail.com, psyu@uic.edu

Abstract—Dialogue Act (DA) classification is a challenging problem in dialogue interpretation, which aims to attach semantic labels to utterances and characterize the speaker’s intention. Currently, many existing approaches formulate the DA classification problem ranging from multi-classification to structured prediction, which suffer from two limitations: a) these methods are either handcrafted feature-based or have limited memories. b) adversarial examples can’t be correctly classified by traditional training methods. To address these issues, in this paper we first cast the problem into a question and answering problem and proposed an improved dynamic memory networks with hierarchical pyramidal utterance encoder. Moreover, we apply adversarial training to train our proposed model. We evaluate our model on two public datasets, i.e., Switchboard dialogue act corpus and the MapTask corpus. Extensive experiments show that our proposed model is not only robust, but also achieves better performance when compared with some state-of-the-art baselines.

Index Terms—dialog act classification, dynamic memory network, adversarial training

I. INTRODUCTION

Dialogue Act (DA) which represents the meaning of utterances has been widely adopted in computational linguistics, especially in the dialogue system. The automatic recognition of DAs is an important step toward understanding spontaneous dialogue, which will facilitate many applications such as human-computer dialogue systems [1], language understanding applications [2], spoken language translation [3], or automatic speech recognition [4]. Table I shows a snippet of dialog with utterances and their corresponding labels. In dialogs, each utterance is assigned a unique DA label, drawn from a well-defined set. Thus, DAs can be thought of as a tag set that classifies utterances according to a combination of pragmatic, semantic, and syntactic criteria. From Table I, we can also find that knowing the past utterances of dialog can help easing the prediction of the current DA state, thus help to narrow the range of utterance generation topics for the current turn [5]. For instance, the “Greeting” and “Farewell” acts are often followed with another same type utterances, the “Answer” act often responds to the former “Question” type utterance.

Motivation. Currently, there have been many research works focusing on the problem of DA classification, ranging from multi-class classification to structured prediction [4], [6]. In

Speaker	Utterance	DA label
A	Hi, long time no see.	Greeting
B	Hi, how are you?	Greeting
A	What are you doing these days?	Question
B	I’m busy writing my paper.	Answer
A	I heard that the deadline is coming.	Statement
B	Yeah.	Backchannel
A	You need to make a push.	Opinion
B	Sure, that’s why I am so busy now.	Agreement
A	I can’t bother you for too long, goodbye.	Farewell
B	See you later.	Farewell

Table I: A snippet of a conversation sample. Each utterance has related dialogue act label [5].

many previous works, hand-crafted features are created and fed into a multi-class classifier such as SVM and Naive Bayes [4], [6], making the model labor intensive and can not be scaled up well across different datasets. To better learn the representation of utterance, recent studies [7], [8] have applied deep learning based models for the DA recognition task, and have shown promising results. However these previous works mainly suffer the limited memory due to the dynamic characteristic of dialog. In other words, each response in dialog is correlated with the previous utterances. For example, it is evident that during a conversation, the speaker’s intent is influenced by the former utterance such as the previous “Greeting” and “Farewell” examples. This limitation makes these models can’t represent dialogues with large numbers of turns and long utterances. Another limitation of these previous works lies in the training process of deep neural networks. It has been shown that traditional neural models are often vulnerable to *adversarial examples*, which are examples created by making small perturbations to the input [9].

Motivated by these above mentioned issues, in this paper, we propose a unified framework integrating dynamic memory network and adversarial learning for DA classification. To better represent the state of utterances, we draw some insights from Dynamic Memory Network (DMN) which have been successfully applied in question and answering [10]. Unlike previous attention-based deep neural networks, DMN can compute dynamic sentence representations dependently and hence can be easily be used for representation of dialog. Specifically, we first formulate the DA classification task into a questioning and answering setting (*Q: What is the label of this*

^{*}Equal contribution

utterance?). Then we propose an improved DMN to learn and memorize the utterance, which contains a hierarchical pyramid utterance encoder. Moreover, we train our model via adversarial training which is a process of training a model to correctly classify both unmodified examples and adversarial examples. It improves not only robustness to adversarial examples, but also generalization performance for original examples. The main contributions of this paper can be summarized as follows:

- Unlike previous studies, to the best of knowledge, it's the first time that we model the DA classification problem from the perspective of question and answering, and propose an improved dynamic memory network with hierarchical pyramid utterance encoder for better representation of utterances.
- To increase the robustness and generalization of our model, we make small perturbations to the input data, and apply adversarial training to train our proposed model.
- Extensive experiments and analysis on two real-world datasets verify the effectiveness of our proposed model when compared with some state-of-the-art baselines.

Organization. The rest of this paper is organized as follows. In Section II, we provide a brief review of the related work about dialogue act recognition problem. In Section III, we first formulate the problem of dialogue act classification from the viewpoint of question and answering, and introduce some background knowledge about dynamic memory network. We elaborate our proposed improved dynamic memory network for DA in Section IV. Extensive experimental results and analysis are presented in Section V. Finally, we provide some concluding remarks in Section VI.

II. RELATED WORK

In this section, we briefly review some related works on dialogue act classification, dynamic memory network and adversarial learning.

A. DA Classification

Most of the existing work for the problem of DA classification can be categorized as following two classes: a) Regarding the DA classification as a multi-classification problem [11], [12]. b) Regarding the DA classification as a sequence labeling problem [13]. Recently, approaches based on deep learning methods improve many state-of-the-art techniques in NLP including DA classification accuracy on open-domain conversations [7], [8], [14], [15]. Kalchbrenner et al. [14] use a mixture of CNN and RNN to represent utterances where CNNs are used to extract local features from each utterance and RNNs are used to create a general view of the whole dialogue. Khanpour et al. [7] design a deep neural network model that benefits from pre-trained word embeddings combined with a variation of the RNN structure for the DA classification task. Ji et al. [8] also investigate the performance of using standard RNN and CNN on DA classification and get the cutting edge results on the MRDA corpus using CNN. Lee et al. [15] propose a model based on CNNs and RNNs that incorporates preceding short texts as context to classify current DAs. Unlike previous

models, we cast the DA classification task into a question and answering problem.

B. Dynamic Memory Network

One line of research related to dynamic memory network is attention and memory mechanism [16], [17], which have been successfully applied in many tasks such as text generation [18], [19] and question answering [20], [21]. In these works, memory is encoded as a continuous representation and operations on memory (e.g. reading and writing) are typically implemented with neural networks. Attention mechanism could be viewed as a compositional function, where lower level representations are regarded as the memory, and the function is to assign a weight to each lower position when computing an upper level representation. Such attention based approaches have achieved promising performances on a variety of NLP tasks [22]. Based on these works, [23] develops dynamic memory network which simultaneously contains memory updating mechanism and attention mechanism. [10] proposes an improved dynamic memory network with some modifications in memory and input module. The dynamic memory network has been successfully applied in many scenarios such as question answering and sentiment analysis. To the best of our knowledge, it's the first time that we apply dynamic memory network in DA classification.

C. Adversarial Learning

Adversarial training [24] introduces an end-to-end and deterministic way of data perturbation by utilizing the gradient information. It could design adversarial examples to attack [25] or improve robustness [26], [27] of neural network models. Adversarial training is originally used in the context of image classification tasks where the input data is continuous. Miyato et al. [9] adopt adversarial training to text classification by adding perturbations on word embeddings and also extends it to a semi-supervised setting by minimizing the entropy of the predicted label distributions on unlabeled data. Wu et al. [28] apply adversarial training in relation extraction within the multi-instance multi-label learning framework. There are also other works on regularizing classifiers by adding random noise to the data, such as dropout [29] and its variant for NLP tasks such as word dropout [30]. In [31], Xie et al. discuss various data noising techniques for language models. Sogaard and Li et al. [32] focus on linguistic adversaries. Inspired by these works, in this paper we apply adversarial training to train our dynamic memory network for the task of DA classification.

III. PRELIMINARIES

In this section, we first declare some notifications through this paper and formulate the DA classification problem mathematically. We also present some background knowledge on dynamic memory network.

A. Task Description

Assume that we have a set \mathcal{D} of N conversations, i.e. $\mathcal{D} = \{C^1, C^2, \dots, C^N\}$ with (Y^1, Y^2, \dots, Y^N) corresponding target DAs. For each conversation C^i , it consists of a series

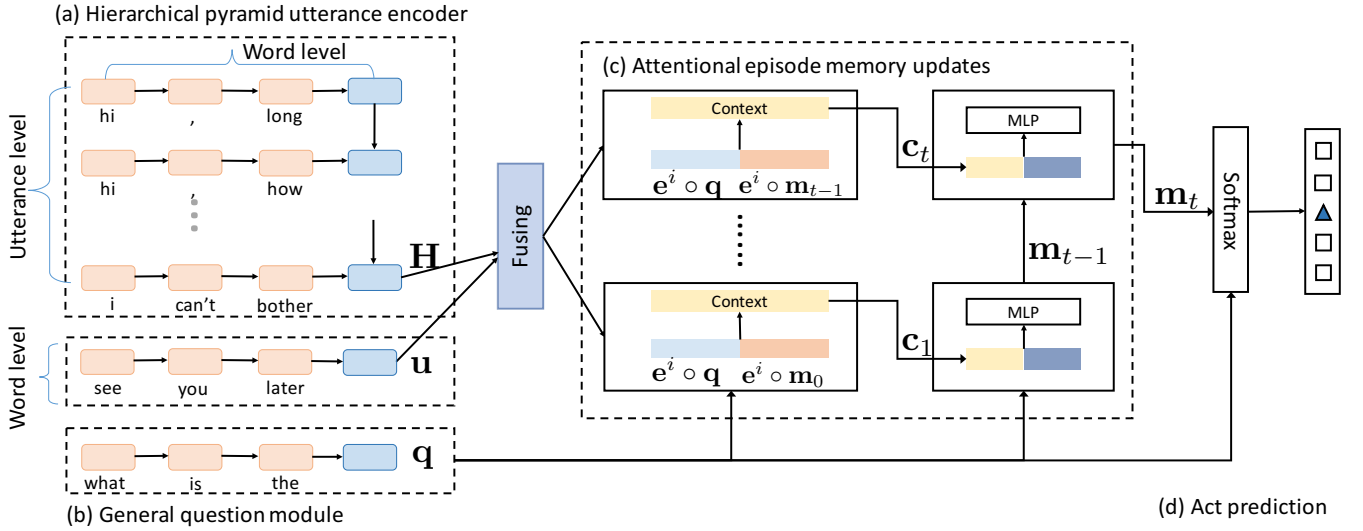


Figure 1: An overview of the network architecture of our proposed model. (a) Hierarchical pyramidal utterance encoder module. (b) General question module. (c) Attentional episode memory updates module. (d) Act prediction module.

of utterances, i.e. $C^i = \{u_1, u_2, \dots, u_d\}$ and the corresponding labels of C^i are $Y^i = \{y_1, y_2, \dots, y_d\}$, where d is the number of utterances. In other words, for each utterance u_j in each conversation, we have an associated target label $y_j \in \mathcal{Y}$, where \mathcal{Y} is the set of all possible DAs. Each utterance u_j in turn is itself a sequence of $|u_j|$ words stringed together, i.e., $u_j = (w_1, w_2, \dots, w_{|u_j|})$.

In this paper, we cast the DA classification problem into a question and answering setting. Specifically, we first fix the question as “what is the label of this utterance?”. Then for a given utterance, our goal is to generate the answer/label for it. For a given conversation $C^i = [u_1, \dots, u_d]$, $u_j = (w_1, w_2, \dots, w_{|u_j|})$ represents an utterance and w_n denotes the n -th word in u_j . Let $H = [u_1, \dots, u_{k-1}]$ denote a dialogue context, the $k-1$ historical utterances, and u_k be a response which means the next utterance. Our goal is to predict the DA label of u_k , given its dialogue context H .

B. Dynamic Memory Networks

Dynamic Memory Network (DMN) is a new neural network architecture based on attention mechanism with the ability of memorizing and reasoning. It has been widely used in question answering tasks since it was first proposed in [23]. In this subsection, we introduce some background knowledge on DMN as presented in [23]. The DMN consists of four modules: input module, question module, episode memory module and answer module.

1) *Input Module*: In this module, the input data corresponding to the question being asked to are encoded into a sequence of distributed vectors representations. We name the vectors as facts (evidences), denoted as $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{|E|}]$, where $|E|$ is the total number of facts, usually is the number of sentence in a document. So \mathbf{e}_i is the vector representation of the i -th sentence. The order of vectors in E cannot be changed at random because memory updates need to be based on the

order of the facts. Long Short-Term memory (LSTM) [33] and Gated Recurrent neural Units (GRU) [34] are typically used to encode the input data. In [23], GRU is selected as an encoder for the trade-off between computational efficiency and performance effectiveness. The detailed operation of GRU is defined as follows:

$$\begin{aligned}
 \mathbf{r}_t &= \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{W}_r \mathbf{g}_{t-1} + \mathbf{b}_r), \\
 \mathbf{z}_t &= \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{W}_z \mathbf{g}_{t-1} + \mathbf{b}_z), \\
 \hat{\mathbf{g}}_t &= \tanh(\mathbf{W}_g \mathbf{x}_t + \mathbf{W}_g (\mathbf{r}_t \circ \mathbf{g}_t) + \mathbf{b}_g), \\
 \mathbf{g}_t &= \mathbf{z}_t \circ \hat{\mathbf{g}}_t + (1 - \mathbf{z}_t) \circ \mathbf{g}_{t-1},
 \end{aligned} \tag{1}$$

where \mathbf{r}_t and \mathbf{z}_t are the reset gate vector and update gate vector respectively, \mathbf{g}_t is the output vector, \mathbf{W} s and \mathbf{b} s are weights matrices and biases, σ is the sigmoid activation function, \circ is an element-wise multiplication.

2) *Question Module*: This module is similar to the input module, which also utilizes GRU as the encoder and encodes a sequence of question words into a distributed vector representation \mathbf{q} . And then \mathbf{q} is fed into episode memory module and answer module separately.

3) *Episode Memory Module*: Episode memory module is the main component of the DMN. It is comprised of an attention mechanism and a memory updating mechanism. This module iterates over the input evidences representations and extracts the information to answer the question \mathbf{q} . When the questions are too complex to answer so that we need reasoning, the episode memory network may iterates over the input evidences representations multiple times. The episode memory may be updated after each iteration. We denote the memory after i -th iteration over the input evidences representations as \mathbf{m}_i . We initialize the foremost \mathbf{m}_0 as the question vector \mathbf{q} . The attention mechanism is in charge of producing a contextual vector \mathbf{c}_i , a weighted sum of the input evidences representation, with relevance of the question \mathbf{q} and the previous memory

\mathbf{m}_{i-1} . The weights of evidences that contribute more to the answer are larger. The episode module may focus on the important information via soft alignment. The memory updating mechanism is in charge of producing the \mathbf{m}_i based on the contextual vector \mathbf{c}_i and the previous memory \mathbf{m}_{i-1} .

4) *Answer Module*: The answer module generates an appropriate answer given the question representation \mathbf{q} and the final episode memory. After multiple updates, \mathbf{m}_T contains all information that is required to answer the question. This module takes different execution modes according to different types of tasks. For single-token-answer case, it is treated as a simple classification task. The token-generation layer is composed of a linear layer with a softmax activation classifier to compute the probability distribution of the answer over the entire vocabulary table. For task that requires generating a sequence of tokens, another GRU is used to decode the concatenation of \mathbf{q} and \mathbf{m}_T to a sequence of tokens.

C. An Overview

Figure 1 shows an overview of the network architecture of our proposed model. The framework can be divided into four parts: (a) Hierarchical pyramid utterance encoder module (cf. Sec. IV-A). In this module, we first represent the input utterance via a pyramidal BiGRU. We first embed each word token into a continuous distributed embedding using *Glove* [35], and then we add a perturbation for each word embedding for adversarial training. Then we fed the perturbed embedding to a pyramidal BiGRU. For dialogue history embedding, we apply a pyramidal BiGRU to embed each utterance and apply another utterance level GRU to represent the whole context. We call this encoder as hierarchical pyramidal utterance encoder. (b) General question module (cf. Sec. IV-B). In this module, we fix the question as “*What is the label of this utterance?*”. We adopt a vanilla BiGRU to embed the question. (c) Attentional episode memory updates module (cf. Sec. IV-C). This module is the core component of our proposed framework which is comprised of attention and memory mechanism. The attention mechanism is to associate a weight to each hidden states (also called facts in DMN). The generated context vector contains the information of facts, question and memory. Then the context vector is fed into the memory updates module for memory updating. (d) Act prediction module (cf. Sec. IV-D). This module is the output module of our proposed framework. In our scenario, we use the softmax layer to map the hidden states to the DA label space. We will elaborate each component of this framework in the following sections.

IV. DMN WITH ADVERSARIAL TRAINING

In this section, we describe our proposed Dynamic Memory Network with Adversarial Learning (ALDMN) for dialogue act classification. We first describe our dynamic memory network with hierarchical pyramidal utterance encoder and general question module. Then, we describe adversarial training to train our model in detail.

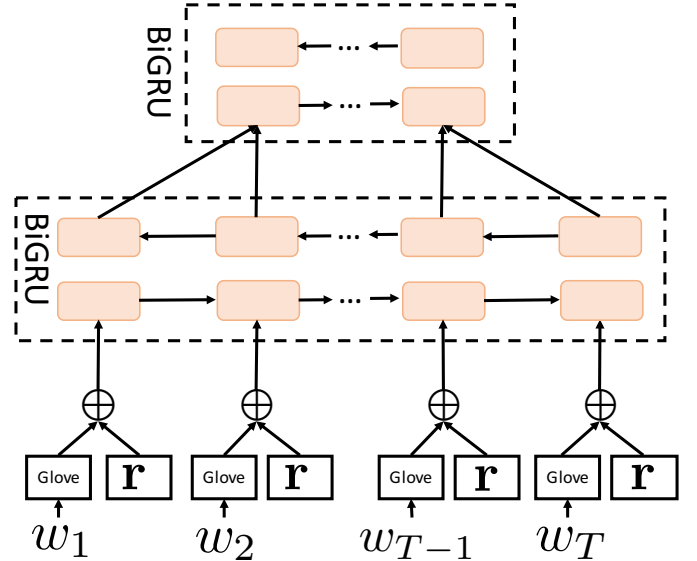


Figure 2: Network structure of pyramidal utterance encoder.

A. Hierarchical Pyramidal Utterance Encoder

In DA classification task, the input is an utterance of sequence of n words $\{w_1, w_2, \dots, w_n\}$ with its history utterances $H = [u_1, \dots, u_{k-1}]$. Therefore, the facts of input are composed of two parts (i.e. input utterance facts \mathbf{e}_u and history utterance facts \mathbf{e}_H). We model these two parts one by one.

To encode each utterance in the dialog, there have been many options such as LSTM [36], Gated Recurrent Unit (GRU) [10], [34]. Pyramidal bidirectional GRU which is an alternative bidirectional GRU that reduces the time dimension after each layer has been successfully applied in character-level sentence encoding and decoding [37]. In this paper we try some available encoders, including LSTM, GRU, pyramidal GRU and their bidirectional versions. Experimental results show that the pyramidal bidirectional GRU achieves the best performance, so we adopt it for utterance encoding.

To start with, for the following adversarial training, since the utterance tokens are discrete, we define the perturbation on continuous word embeddings instead of discrete word inputs. Thus, we first use *Glove* [35] to transform an utterance of n words $\{w_1, w_2, \dots, w_n\}$ into distributed continuous embeddings, and then add perturbations to them. The word vector of w_i can be represented as follows:

$$\mathbf{v}(w_i) = \text{Glove}(w_i) + \mathbf{r}_{adv}. \quad (2)$$

We then feed the word embeddings into a pyramidal bidirectional GRU. The hidden states in each bidirectional GRU are defined as follows:

$$\vec{\mathbf{f}}_t^i = \text{GRU}(\vec{\mathbf{f}}_{t-1}^i, \mathbf{e}_t^{i-1}), \quad (3)$$

$$\overleftarrow{\mathbf{f}}_t^i = \text{GRU}(\overleftarrow{\mathbf{f}}_{t+1}^i, \mathbf{e}_t^{i-1}), \quad (4)$$

$$\mathbf{h}_t^i = \vec{\mathbf{f}}_t^i + \overleftarrow{\mathbf{f}}_{t+1}^i, \quad (5)$$

where GRU denotes the gated recurrent unit function, which has shown to improve the performance of RNNs [38], [39].

Input utterance facts. For pyramidal Bi-GRU, the input from the previous layer input $\mathbf{e}_t^0 = \mathbf{v}_t^0$ and

$$\mathbf{e}_{u,t}^i = \tanh(\mathbf{W}_{pyr}^{(i)}[\mathbf{h}_{2t}^{i-1}; \mathbf{h}_{2t+1}^{i-1}]) + \mathbf{b}_{pyr}^{(i)}, \quad (6)$$

for $i > 0$. The weight matrix $\mathbf{W}_{pyr}^{(i)}$ thus reduces the number of hidden states for each additional hidden layer by half, and hence the encoder has a pyramidal structure. At the final hidden layer we obtain the encoded representation \mathbf{e} consisting of $\lceil T/2^{N-1} \rceil$ hidden states, where N denotes the number of hidden layers. We can find that after the input sequence passes through the pyramidal encoder, the number of encoder output becomes less, which reduces the computational load for the following attention mechanism.

Historical utterance facts. After encoding each utterance via pyramidal Bi-GRU, we apply another GRU to represent the historical utterances:

$$\mathbf{e}_{H,t}^i = \text{GRU}(\mathbf{e}_{H,t-1}^i, \mathbf{e}_{u,t-1}^{i-1}). \quad (7)$$

Finally, we can denote the input facts by $\mathbf{e} = \{\mathbf{e}_u^1, \dots, \mathbf{e}_u^{|u|}, \dots, \mathbf{e}_H^1, \dots, \mathbf{e}_H^{|H|}\}$.

B. General Question Module

Different from the question answering task, in DA classification task, none of the utterance is provided with a question. Actually, we can consider all of the utterances facing the same question, such as “What is the label of this utterance?”. Thus, we compute a general vector \mathbf{q} representing the same question, where \mathbf{q} is jointly learned with other model parameters and used in the subsequent module.

C. Attentional Episode Memory Updates

Attention mechanism and memory updating mechanism constitute the main components of this module. The episodic memory module, as depicted in Figure 1 (c), retrieves information from the input facts $\mathbf{e} = \{\mathbf{e}_u^1, \dots, \mathbf{e}_u^{|u|}, \dots, \mathbf{e}_H^1, \dots, \mathbf{e}_H^{|H|}\}$ provided to it by focusing attention on a subset of these facts. We implement this attention by associating a single scalar value, the attention gate α_t^i , with each fact \mathbf{e}^i during pass t .

Here the implementation of attention mechanism is the same as [17]. At the t -th iteration, we concatenate the output \mathbf{e} of the pyramidal encoder with previous iteration episodic memory \mathbf{m}_{t-1} and question vector \mathbf{q} , and then employ the basic soft attention to obtain the t -th contextual vector as:

$$\mathbf{z}_t^i = [\mathbf{e}^i \circ \mathbf{q}; \mathbf{e}^i \circ \mathbf{m}_{t-1}], \quad (8)$$

$$\beta_t^i = \mathbf{W}^{(2)} \tanh(\mathbf{W}^{(1)} \mathbf{z}_t^i + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)}, \quad (9)$$

$$\alpha_t^i = \frac{\exp(\beta_t^i)}{\sum_{j=1}^n \exp(\beta_t^j)}, \quad (10)$$

$$\mathbf{c}_t = \sum_{i=1}^n \alpha_t^i \mathbf{e}^i, \quad (11)$$

where \circ is the element-wise multiplication, $[\cdot; \cdot]$ is the concatenation operation, \mathbf{W} and \mathbf{b} are model parameters. In [10], \mathbf{z}_t^i is

set to $\mathbf{z}_t^i = [\mathbf{e}^i \circ \mathbf{q}; \mathbf{e}^i \circ \mathbf{m}_{t-1}; |\mathbf{e}^i - \mathbf{q}|; |\mathbf{e}^i - \mathbf{m}_{t-1}|]$, where $|\cdot|$ is the element-wise absolute value. However in our scenario, $\mathbf{z}_t^i = [\mathbf{e}^i \circ \mathbf{q}; \mathbf{e}^i \circ \mathbf{m}_{t-1}]$ achieves better performance from the experimental results.

Following the memory update component used in [40], we first concatenate the previous episodic memory \mathbf{m}_{t-1} , the current contextual vector \mathbf{c}_t and question vector \mathbf{q} , and use a ReLU layer for the memory update:

$$\mathbf{m}_t = \text{ReLU}(\mathbf{W}^{(3)}[\mathbf{m}_{t-1}; \mathbf{c}_t; \mathbf{q}] + \mathbf{b}^{(3)}), \quad (12)$$

where $[\cdot; \cdot]$ is the concatenation operation, $\mathbf{W}^{(3)}$ and $\mathbf{b}^{(3)}$ are model parameters. We use ReLU activation and initialize the memory vector as the question vector: $\mathbf{m}_0 = \mathbf{q}$.

D. Act Prediction

The final episodic memory \mathbf{m}_T and question vector \mathbf{q} are concatenated to compute the probability of the correct answer. In DA classification, the answer is the label of the corresponding utterance, which is different from the question answering task predicting a sequence token representing the correct answer. The classifier takes the episodic memory \mathbf{m}_T and question vector \mathbf{q} as input:

$$p(y|\mathbf{m}_T, \mathbf{q}) = \text{softmax}(\mathbf{W}^{(4)}[\mathbf{q}; \mathbf{m}_T] + \mathbf{b}^{(4)}), \quad (13)$$

where y represents the index of multi-class candidate answer. $\mathbf{W}^{(4)}$ and $\mathbf{b}^{(4)}$ are learnable parameters to be optimized in the model.

The loss function is the negative log-likelihood of the true class labels y for each utterance:

$$\mathcal{L}(\mathbf{u}, \theta) = - \sum_{i=1}^M \log p(y^{(i)}|\mathbf{u}; \theta), \quad (14)$$

where M is the number of utterances in training dataset.

E. Adversarial Training

For classification task, many existing models use dropout or ℓ_2 for model regularization and achieve a good performance. However, these models demonstrate poor performance on adversarial examples even sometimes the perturbations are so small that humans cannot detect it. Adversarial learning is another regularization method of the model, which aims to train the model to correctly classify adversarial examples and real examples. Following the method of adding perturbations in [9], we create continuous perturbations by adding noise to word embedding. Let \mathbf{u} denote the input utterance and θ the parameters of model. When applied to a model, adversarial training adds the following term to the loss function:

$$\mathcal{L}_{adv}(\theta) = - \sum_{i=1}^M \log p(y_i|\mathbf{u}_i + \mathbf{r}_{adv,i}; \theta), \quad (15)$$

where M is the total number of utterances in training data, \mathbf{r}_{adv} is the worst case perturbations against the current model which can be defined as follows:

$$\mathbf{r}_{adv} = \underset{\mathbf{r}, \|\mathbf{r}\| < \epsilon}{\text{argmin}} \log p(y|\mathbf{u} + \mathbf{r}; \hat{\theta}), \quad (16)$$

Dataset	$ C $	$ V $	MinL	MeanL	MaxL
SwDA	42	19K	16	56	118
MapTask	13	15K	2	4	21

Table II: $|C|$ is the number of Dialogue Act classes, $|V|$ is the vocabulary size. MinL, MeanL and MaxL indicate the minimum, mean and maximum of utterance length, respectively.

where \mathbf{r} is a perturbation on the input utterance \mathbf{u} and $\hat{\theta}$ is a constant set to the current parameters of a classifier, indicating that the backpropagation algorithm will be used to propagate gradients through the adversarial example construction process.

We can see that the \mathbf{r}_{adv} can not be calculated exactly in general, since exact minimization with respect to \mathbf{r} is intractable for many models such as neural networks. Inspired by [24], we can approximate this value by linearizing $\log(y|\mathbf{u}; \theta)$ around \mathbf{u} . Therefore, with a linear approximation and a ℓ_2 norm constraint in Eq.16, the resulting adversarial perturbation is formulated as follows:

$$\mathbf{r}_{adv} = \frac{\epsilon \mathbf{g}}{\|\mathbf{g}\|}, \quad (17)$$

where \mathbf{g} is the normalized gradient of log-likelihood with respect to \mathbf{u} , which can be formulated as follows:

$$\mathbf{g} = \nabla_{\mathbf{u}} \log p(y|\mathbf{u}; \hat{\theta}). \quad (18)$$

We can compute the perturbation by backpropagation in neural networks. To optimize the objective, we employ the Adam [41], an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments.

V. EXPERIMENTS AND ANALYSIS

In this section, we conduct extensive experiments and analysis on two real-world datasets to show the effectiveness of our proposed approach.

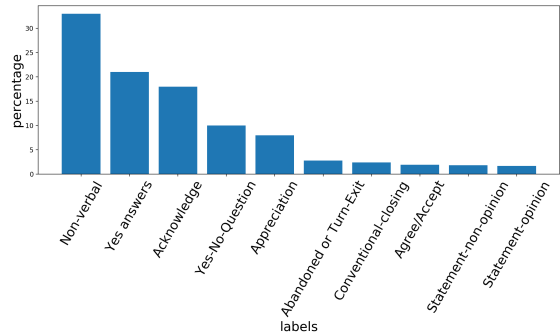
A. Datasets

We evaluate the performance of our model on two benchmark datasets used in several prior studies for the DA classification task, i.e. Switchboard Dialogue Act Corpus (SwDA) [42] and MapTask Dialogue Act Corpus (MapTask) [43].

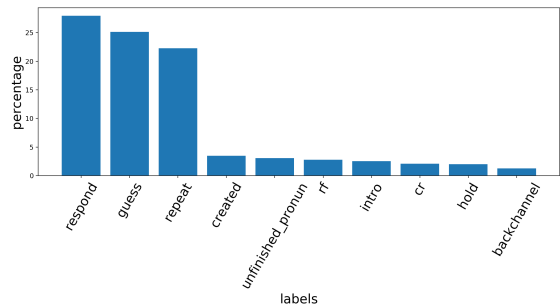
SwDA: This dataset consists of 1,155 telephone conversations and each of the utterance in the dialogues is mapped into 42 distinguished utterance types via DAMSL taxonomy. We shuffle the data randomly and use 1,050 conversations for training and 105 conversations for testing.

MapTask: This dataset comprises of 128 dialogues and more than 27,000 utterances. Each of the utterance is labeled with one of the 13 tags. Unlike SwDA, the MapTask corpus emphasizes on directions and conductions.

Table II presents different statistics for both datasets. For SwDA, training and testing sets are provided but not the validation set, so we use the standard practice of taking a part of training data set as validation set [15]. We also shows the top 10 utterance labels on different datasets in Figure 3.



(a) SwDA



(b) MapTask

Figure 3: The distribution of utterance labels on two datasets: a) SwDA, b) MapTask.

B. Evaluation Metrics

We employ the standard accuracy as the metric of evaluating our proposed ALDMN method. The accuracy is defined as:

$$Accuracy = \frac{1}{M} \sum_{i=1}^M \mathbf{1}[\hat{y}_i = y_i], \quad (19)$$

where \hat{y}_i and y_i are the predicted label and ground true label, respectively. $\mathbf{1}[\cdot]$ is the indicator function. When the predicted label is the same as the true, label $\mathbf{1}[\cdot]$ equals to 1 otherwise 0.

C. Implementation Details

To train our proposed model, we first randomize the training data and set the mini-batch to 128. For each batch, the utterance is padded with a special token $\langle \text{pad} \rangle$ to the maximum length. We regard the words whose occurrence is less than 2 as Out of Vocabulary (OOV) tokens and replace them with $\langle \text{unk} \rangle$ s. We remove all punctuation marks except interrogation and all characters are converted to lower-case. We set the memory updating iterations to 3. We adopt Adam [41] optimizer with the learning rate initialized to 0.01. We run the training data set for 45 epochs with early stopping when validation loss did not decline for five consecutive epochs. The random uniform initialization with range $[-0.1, 0.1]$ is used for all matrix variables, including word embedding and other weights. Both the embedding and hidden dimensions are set to $d = 200$. The

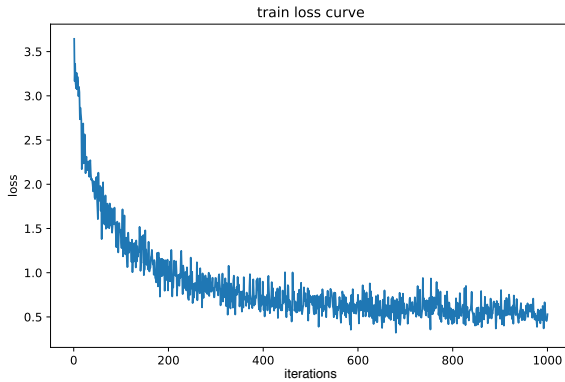


Figure 4: Training loss of each iteration.

dropout rate is set to 0.2. Figure 4 is the training loss curve of our method. We can see that our model is converged after around 300 iterations.

All the experiments in this paper are implemented with Python 2.7 based on TensorFlow, and run on a computer with an 2.2 GHz Intel Core i7 CPU, 64 GB 1600 MHz DDR3 RAM, and a Titan X GPU with 12 GB memory, running Ubuntu 16.04.

D. Results and Analysis

We compare our proposed method with several other state-of-the-art methods for the problem of dialogue act classification as follows:

- UCI [44] method embeds contextual information of utterance via hierarchical CNN/RNN for DA classification.
- PDI [45] method predicts the next label based on the current label probability distribution to avoid label bias.
- DRLM-Conditional [46] method is a latent variable recurrent neural network architecture for jointly modeling utterances and DA labels.
- BiLSTM-Softmax [7] method uses a bidirectional LSTM to embed utterances and then feeds them to a softmax classifier.
- RCNN [14]. Different from BiLSTM-Softmax, this method uses hierarchical CNN to embed utterances.
- DMN [23] methods is a DMN based method for sentiment classification, where BiLSTM is used for sentences embedding.
- ADC [47] and PDI [45] methods are acoustic and discourse classification based on HMM and SVM.
- GAN [48] method is a generative neural network which incorporates attention technique and a label-to-label connection.

Table III and Table IV respectively show the experimental *Accuracy* results of the methods on the SwDA and MRDA datasets. The hyper-parameters and parameters which achieve the best performance on the validation set are chosen to conduct the testing evaluation. From these two tables, we can find that our proposed model ALDMN definitely outperforms other baselines in both datasets. On SwDA dataset, our model

Model	Accuracy(%)
RCNN(Bulnsom et al. 2013)	73.9
BiLSTM-Softmax(Khanpour et al. 2016)	75.8
DRLM-Conditional(Ji et al. 2016)	77.0
PDI(Tran et al. 2017)	75.6
UCI(Liu et al. 2017)	79.9
DMN(Kumar et al., 2015)	75.2
ALDMN (Our Model)	81.5

Table III: Classification accuracy on SwDA corpus, comparing our ALDMN model with other methods as described in literatures.

Model	Accuracy(%)
ADC(Julia et al. 2010)	55.4
GAN(Tran et al. 2017b)	62.9
PDI(Tran et al. 2017a)	65.9
DMN(Kumar et al., 2015)	64.7
ALDMN (Our Model)	68.5

Table IV: Classification accuracy on MapTask corpus, comparing our ALDMN model with other methods as described in literatures.

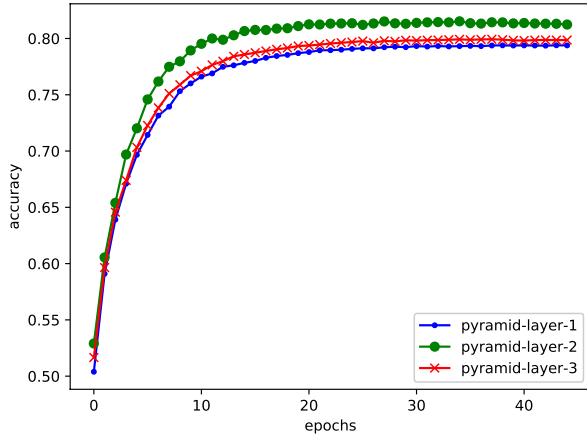
improves the accuracy over the best performing traditional method UCI by 1.6%. On MapTask dataset, our model improves the accuracy over the best performing traditional method PDI by 2.6%. This verifies the effectiveness of our proposed DMN module and adversarial training approach for DA classification.

E. Impact of Pyramidal Encoder

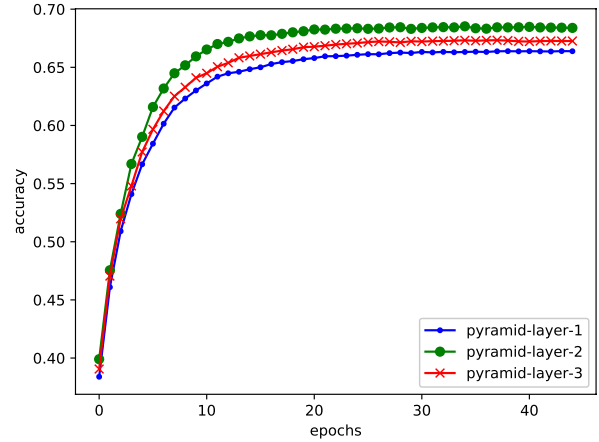
To analyze the impact of pyramidal layer, we vary the layers number of pyramidal encoder from 1 to 3. When the layer number is 1, the pyramidal encoder reduced into a BiGRU encoder. Figure 5 shows the classification accuracy with varying pyramid layers on SwDA and MapTask dataset. From this figure, we can find that our model achieves the best performance when the pyramid layer is set to 2. This verifies the effectiveness of the pyramidal encoder. Moreover, the performance decrease when the pyramid layer is set to 2. This can be illustrated by the fact that our encoder may be over-fitting with the number of layers increasing.

F. Impact of Adversarial Learning

To demonstrate the effectiveness of adversarial learning on word embeddings, we compare architectures with/without adversarial learning. For adversarial training, we set $\epsilon = 3$. For model without adversarial training, the dropout is applied. Comparing performances are shown in Figure 6. From this figure, we can find that our model with adversarial can really improve the accuracy performance in both two datasets. Take SwDA as an example, with only dropout applied, our model achieves accuracy of 79.1%, while the accuracy of model with adversarial learning can reach up to 81.5%. This can be explained by the fact that the model without adversarial learning is strongly affected by the syntax of the dataset. In fact, in DA classification task, as the conversations are sometimes recorded in a casual setting, the grammar of utterances are not so strict compared to a formal document. What's more, due to

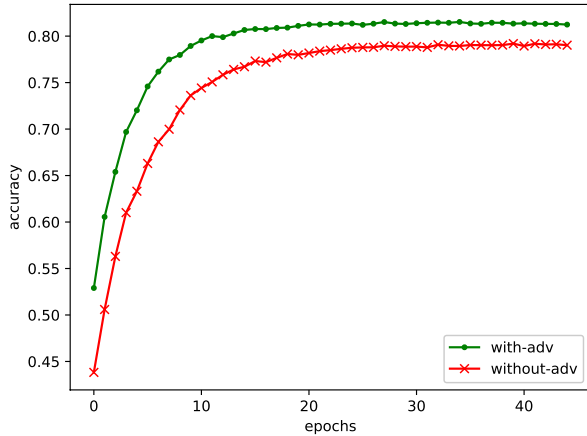


(a) SwDA

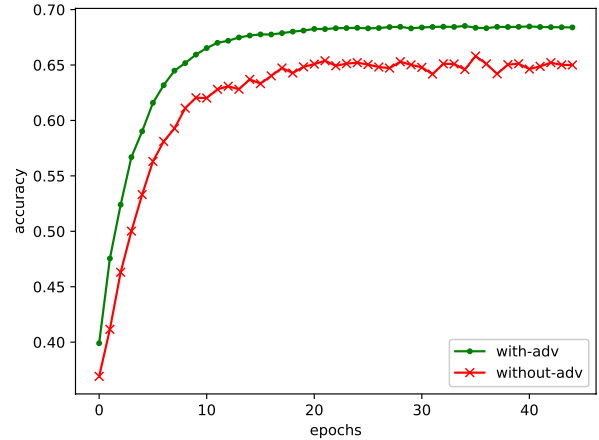


(b) MapTask

Figure 5: Classification accuracy curves with different pyramid layer of pyramid-layer-1, pyramid-layer-2, pyramid-layer-3 on the two different datasets: a) SwDA, b) MapTask.



(a) SwDA



(b) MapTask

Figure 6: Classification accuracy curves of with (green) and without (red) adversarial learning on two different datasets: a) SwDA, b) MapTask.

the model pre-training step, some antonyms, although greatly different in semantics, are very close in the vector space. For example, “big” is opposite to “small” in semantics, but they are sometimes in the similar context. So they are assigned the similar word embeddings on the baseline model while the embeddings cannot convey the actual meaning of these two words. Adversarial learning ensures the representation of sentences not be changed under some perturbations so that those words in same context with different meanings can be separated in the vector space.

G. Performance on Various Data Distribution

We vary the length of utterances since the utterance length may have an effect on the representation. Figure 7 shows the performance of our proposed model w.r.t. varying utterance

on SwDA dataset. From this figure, we can find that our model achieves the best performance on the utterances whose length is around 40. This can be illustrated by the fact that short utterances always contain limited information while long utterances may contain more noises.

H. Confusion Matrix Visualization

Figure 8 shows the confusion matrix of our proposed model for the SwDA dataset. Among them the most confused pairs are (sd, sv) and (aa, b) which represent (statement-non-opinion, statement-opinion) and (agree-accept, acknowledge) respectively. The total number of utterances with DA ‘sd’, ‘sv’, ‘aa’, and ‘b’ are 72,824, 25,197, 10,820 and 37,096, respectively. 3,423 utterances (4.7%) with true label non-opinion are predicted incorrectly as opinion, whereas, 64,847

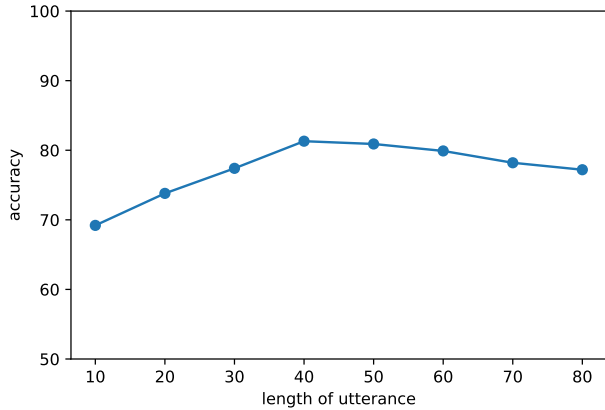


Figure 7: The performance of our proposed model w.r.t. varying length of utterances on SwDA dataset.

fo	61.3	5.9	0.4	0.0	6.8	0.0	0.0	0.0	0.0	
qw	0.0	77.2	0.0	0.0	3.0	1.1	0.0	0.0	0.0	
qyd	0.0	0.0	39.3	26.4	10.2	0.0	0.0	0.0	14.3	
qy	0.0	0.3	0.7	81.9	0.0	0.0	0.0	0.0	2.1	
sd	0.0	0.0	0.4	0.0	89.4	0.2	0.0	0.1	4.0	
ad	0.0	0.0	0.0	0.0	32.1	29.6	0.0	0.0	23.2	
h	0.0	0.0	0.0	0.0	16.3	0.0	64.3	0.0	5.4	
aa	0.0	0.0	0.0	0.0	5.2	0.0	0.0	79.3	12.6	
b	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.8	90.4	
sv	0.0	0.0	0.3	0.0	25.4	0.0	0.0	2.3	0.6	
	fo	qw	qyd	qy	sd	ad	h	aa	b	sv

Figure 8: Confusion matrix of our proposed model for the SwDA dataset, where the row denotes the true label and the column denotes the predicted label. The numbers in the bracket besides the DA label in the rst cell of each row is the count of the number of utterances of that DA label.

utterances (89.7%) with true label non-opinion are predicted correctly. Similarly, 6,478 utterances (25.4%) with true label opinion are predicted incorrectly as non-opinion whereas 17,638 utterances (70.5%) with true label opinion are predicted correctly. On further analysis of the cause of this confusion between these two class pairs, we identify that some utterances are classified correctly by the model. However, they are marked incorrectly classified because of bias in the ground truth. For some of the utterances, classes are not distinguishable even by humans because of the subjectivity.

VI. CONCLUSION

In this paper, we propose a unified framework integrating dynamic memory network and adversarial training for the task of dialogue act classification. Specifically, we first cast the act classification task into a question and answering problem and propose an improved dynamic memory network to represent and memorize the utterance with its corresponding history utterances. In addition, we apply adversarial training to train a more robust model. We demonstrate the effectiveness of our

proposed model using the well-known public datasets SwDA and MapTask. Extensive experiments demonstrate that our model can achieve better performance than several state-of-the-art solutions to the problem.

VII. ACKNOWLEDGEMENT

This work is partially supported by the Ministry of Education of China under grant of No.2017PT18, the Natural Science Foundation of China under grant of No. 61672453, 61773361, 61473273, 61602405, the WE-DOCTOR company under grant of No. 124000-11110 and the Zhejiang University Education Foundation under grant of No. K17-511120-017, No. K17-518051-021. This work is also supported by CCF-Tencent Open Research Fund, NSF through grants IIS-1526499, IIS-1763325, CNS1626432, and NSFC 61672313.

REFERENCES

- [1] D. R. Traum, "Speech acts for dialogue agents," in *Foundations of rational agency*. Springer, 1999, pp. 169–201.
- [2] A. Ezen-Can and K. E. Boyer, "Understanding student language: An unsupervised dialogue act classification approach," *Journal of Educational Data Mining (JEDM)*, vol. 7, no. 1, pp. 51–78, 2015.
- [3] N. Reithinger, R. Engel, M. Kipp, and M. Klesen, "Predicting dialogue acts for a speech-to-speech translation system," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 2. IEEE, 1996, pp. 654–657.
- [4] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. V. Ess-Dykema, and M. Meteer, "Dialogue act modeling for automatic tagging and recognition of conversational speech," *Computational linguistics*, vol. 26, no. 3, pp. 339–373, 2000.
- [5] Z. Chen, R. Yang, Z. Zhao, D. Cai, and X. He, "Dialogue act recognition via crf-attentive structured network," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2018, pp. 225–234.
- [6] S. Grau, E. Sanchis, M. J. Castro, and D. Vilar, "Dialogue act classification using a bayesian approach," in *9th Conference Speech and Computer*, 2004.
- [7] H. Khanpour, N. Guntakandla, and R. Nielsen, "Dialogue act classification in domain-independent conversations using a deep recurrent neural network," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2012–2021.
- [8] Y. Ji, G. Haffari, and J. Eisenstein, "A latent variable recurrent neural network for discourse relation language models," *arXiv preprint arXiv:1603.01913*, 2016.
- [9] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," *arXiv preprint arXiv:1605.07725*, 2016.
- [10] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," *international conference on machine learning*, pp. 2397–2406, 2016.
- [11] N. Reithinger and M. Klesen, "Dialogue act classification using language models," in *Fifth European Conference on Speech Communication and Technology*, 1997.
- [12] J. Geertzen, V. Petukhova, and H. Bunt, "A multidimensional approach to utterance segmentation and dialogue act classification," in *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, Antwerp, 2007, pp. 140–149.
- [13] D. Surendran and G.-A. Levow, "Dialog act tagging with support vector machines and hidden markov models," in *Ninth International Conference on Spoken Language Processing*, 2006.
- [14] N. Kalchbrenner and P. Blunsom, "Recurrent convolutional neural networks for discourse compositionality," *arXiv preprint arXiv:1306.3584*, 2013.
- [15] J. Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," *arXiv preprint arXiv:1603.03827*, 2016.
- [16] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

- [17] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
- [18] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4651–4659.
- [19] Y. Wan, Z. Zhao, M. Yang, G. Xu, H. Ying, J. Wu, and P. S. Yu, "Improving automatic source code summarization via deep reinforcement learning," in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*. ACM, 2018, pp. 397–407.
- [20] Z. Zhao, Q. Yang, D. Cai, X. He, and Y. Zhuang, "Video question answering via hierarchical spatio-temporal attention networks," in *International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 2, 2017, p. 1.
- [21] B. Pan, H. Li, Z. Zhao, B. Cao, D. Cai, and X. He, "Memen: multi-layer embedding with memory networks for machine comprehension," *arXiv preprint arXiv:1707.09098*, 2017.
- [22] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.
- [23] A. Kumar, O. Isroy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in *International Conference on Machine Learning*, 2016, pp. 1378–1387.
- [24] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *international conference on learning representations*, 2015.
- [25] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *ICLR*, 2018.
- [26] J. Wang, J. Zhang, W. Bao, X. Zhu, B. Cao, and P. S. Yu, "Not just privacy: Improving performance of private deep learning in mobile cloud," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 2407–2416.
- [27] J. Zhang, J. Wang, L. He, Z. Li, and P. S. Yu, "Layerwise perturbation-based adversarial training for hard drive health degree prediction," *International Conference on Data Mining*, 2018.
- [28] Y. Wu, D. Bamman, and S. Russell, "Adversarial training for relation extraction," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1778–1783.
- [29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [30] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III, "Deep unordered composition rivals syntactic methods for text classification," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2015, pp. 1681–1691.
- [31] Z. Xie, S. I. Wang, J. Li, D. Lévy, A. Nie, D. Jurafsky, and A. Y. Ng, "Data noising as smoothing in neural network language models," *arXiv preprint arXiv:1703.02573*, 2017.
- [32] Y. Li, T. Cohn, and T. Baldwin, "Robust training under linguistic adversity," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, vol. 2, 2017, pp. 21–27.
- [33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: <http://dx.doi.org/10.1162/neco.1997.9.8.1735>
- [34] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv: Neural and Evolutionary Computing*, 2014.
- [35] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
- [36] H. Kumar, A. Agarwal, R. Dasgupta, S. Joshi, and A. Kumar, "Dialogue act sequence labeling using hierarchical encoder with crf," *arXiv preprint arXiv:1709.04250*, 2017.
- [37] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4960–4964.
- [38] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.
- [39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [40] B. Peng, Z. Lu, H. Li, and K.-F. Wong, "Towards neural network-based reasoning," *arXiv preprint arXiv:1508.05508*, 2015.
- [41] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [42] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, vol. 1. IEEE, 1992, pp. 517–520.
- [43] A. H. Anderson, M. Bader, E. G. Bard, E. Boyle, G. Doherty, S. Garrod, S. Isard, J. Kowtko, J. McAllister, J. Miller *et al.*, "The hcrc map task corpus," *Language and speech*, vol. 34, no. 4, pp. 351–366, 1991.
- [44] Y. Liu, K. Han, Z. Tan, and Y. Lei, "Using context information for dialog act classification in dnn framework," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2170–2178.
- [45] Q. H. Tran, I. Zukerman, and G. Haffari, "Preserving distributional information in dialogue act classification," *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2151–2156, 2017.
- [46] Y. Ji, G. Haffari, and J. Eisenstein, "A latent variable recurrent neural network for discourse relation language models," *north american chapter of the association for computational linguistics*, pp. 332–342, 2016.
- [47] F. N. Julia, K. M. Iftekharuddin, and A. U. ISLAM, "Dialog act classification using acoustic and discourse information of maptask data," *International Journal of Computational Intelligence and Applications*, vol. 9, no. 04, pp. 289–311, 2010.
- [48] Q. H. Tran, G. Haffari, and I. Zukerman, "A generative attentional neural network model for dialogue act classification," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, 2017, pp. 524–529.