Deep Contextualized Utterance Representations for Response Selection and Dialogue Analysis

Jia-Chen Gu, TianDa Li, Zhen-Hua Ling, Senior Member, IEEE, Quan Liu, Zhiming Su, Yu-Ping Ruan, and Xiaodan Zhu

Abstract—The NOESIS II challenge, as the Track 2 in the Eighth Dialogue System Technology Challenge (DSTC 8), is the extension of Track 1 in DSTC 7. Three new elements are incorporated into the extended track, i.e., dialogue with multiple participants, dialogue success, and dialogue disentanglement. These are vital for the creation of a deployed task-oriented dialogue system. This track is divided into four subtasks, the first two of which are evaluated in the form of response selection and the last two focus on dialogue analysis. This paper describes our methods developed for these four subtasks, which all employ deep contextualized utterance representations to make models aware of contextual information and to keep the intrinsic property of multi-turn dialogue systems. In the released evaluation results of Track 2 in DSTC 8, our proposed methods ranked fourth in subtask 1, third in subtask 2, and first in subtask 3 and subtask 4 respectively. In addition to the challenge tasks, we also compare our proposed methods with previous ones on public benchmark datasets. Experimental results show that our proposed methods outperform existing ones by large margins and achieve new state-of-the-art performances on multi-turn response selection and dialogue disentanglement.

Index Terms—dialogue system technology challenge, response selection, multiple participants, dialogue success, dialogue disentanglement, deep contextualized utterance representations.

I. INTRODUCTION

Enabling dialogue systems to converse naturally with humans is a challenging yet intriguing problem of artificial intelligence. Recently, human-computer conversation has attracted increasing attention due to its promising potentials and alluring commercial values. Many dialogue assistants such as Apple Siri, Google Now and Microsoft Cortana emerge and become popular. These dialogue systems aim to engage users in human-computer conversations in the open domain.

This article was presented in part at the Eighth Dialogue System Technology Challenge (DSTC 8) Workshop, New York, USA, Jan. 2020 [1]. The work on evaluating our proposed methods for subtask 1 and subtask 2 on public benchmarks has been accepted as a short paper by the 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, Oct. 2020 [2]. This work was supported by National Key RD Program of China (Grant No. 2019YFF0303001), and the National Nature Science Foundation of China (Grant No. 61871358). The last author’s research is supported by NSERC Discovery Grants and Discovery Accelerator Supplement Grants (DAS).

J.-C. Gu, Z.-H. Ling, Q. Liu and Y.-P. Ruan are with the National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China, Hefei 230027, China (e-mail: gujc@mail.ustc.edu.cn; quanliu@ustc.edu.cn; ypruan@mail.ustc.edu.cn).

T. Li and X. Zhu are with the Department of Electrical and Computer Engineering and Ingenuity Labs, Queen’s University, Kingston, Canada (tianda.li@queensu.ca; xiaodan.zhu@queensu.ca).

Q. Liu and Z. Su are with the State Key Laboratory of Cognitive Intelligence, iFLYTEK Company Ltd., Hefei 230088, China (e-mail: quanliu@iflytek.com; zmsu@iflytek.com).

Building on the success of Track 1 in the Seventh Dialogue System Technology Challenge (DSTC 7) (NOESIS: Noetic End-to-End Response Selection Challenge) [3], an extension as Track 2 in DSTC 8 (NOESIS II: Predicting Responses, Identifying Success, and Managing Complexity in Task-Oriented Dialogue) has been proposed [4]. Its basic task is multi-turn response selection in retrieval-based chatbots [2], [5]–[16]. Furthermore, new elements that are vital for the creation of a deployed task-oriented dialogue system are incorporated, including (1) dialogue with multiple participants, (2) predicting whether and where a dialogue has solved the problem yet, and (3) handling multiple simultaneous dialogues. Each of these adds a new dimension and brings the track closer to building practical systems.

In this paper, we describe our methods developed for the four subtasks of Track 2 in DSTC 8. Recently, deep contextualized representations have been popularly studied in the domain of natural language processing [17]–[19]. They capture rich language information from texts and show great performance on many downstream tasks, such as natural language inference (NLI) [20] and question answering (QA) [21]. Thus, deep contextualized representations are also employed for building our models in this paper. Bidirectional Encoder Representations from Transformers (BERT) [18], a pre-trained language model, is adopted as the basic of our work. Furthermore, several methods are proposed to make the contextualized representations compatible with each subtask respectively. For subtask 1, first, in order to make pre-trained language models aware of the speaker change information during the conversation, we propose to enhance the model by adding speaker embeddings to token representation and adding special segmentation tokens between context utterances. Second, domain adaptation is performed to incorporate specific in-domain knowledge into pre-trained language models. These two strategies are designed to enhance the contextual information and to improve the conversation understanding capability of a multi-turn dialogue system. For subtask 2, in order to tackle the entangled dialogue which are mixed with multiple conversation topics and are composed of hundreds of utterances, a heuristic speaker-aware disentanglement strategy is proposed, which helps to select a small number of most important utterances according to the speaker information in them. For subtask 3, we formulate the problem as a combination of sequence labeling and natural language inference. Each utterance is considered as a unit when performing sequence labeling. Furthermore, natural language inference is performed for each unit. Most importantly, a context-level RNN is employed to aggregate
the contextual information across the whole dialogue, which returns the deep contextualized utterance representations for inference. For subtask 4, a hierarchical BERT-based model is proposed to identify the utterances occurring in the same conversation section. The BERT model is used to calculate the similarity between utterances. Similarly, a context-level RNN is employed on top of the output of BERT in order to aggregate the contextual semantics across different messages and return the deep contextualized utterance representations. As shown in the released evaluation results on the unseen test sets, our proposed methods ranked fourth in subtask 1, third in subtask 2, and first in subtask 3 and subtask 4 respectively.

This paper is an extended version of our paper [1] that has been presented at the DSTC 8 workshop. First, in order to compare our method proposed for subtask 1 with previous ones, we further evaluate it on four large-scale public multi-turn response selection datasets, i.e., Ubuntu Dialogue Corpus V1 [5], Ubuntu Dialogue Corpus V2 [6], Douban Conversation Corpus [7] and E-commerce Dialogue Corpus [8]. Experimental results show that our proposed method outperforms the existing ones on all metrics by large margins. Specifically, the margins are 5.5% \( R_{10}@1 \) on Ubuntu Dialogue Corpus V1, 5.9% \( R_{10}@1 \) on Ubuntu Dialogue Corpus V2, 3.2% MAP and 2.7% MRR on Douban Conversation Corpus, and 8.3% \( R_{10}@1 \) on E-commerce Corpus, leading to new state-of-the-art performances on multi-turn response selection. Second, we compare our method proposed for subtask 4 with previous ones on the benchmark of dialogue disentanglement, i.e., Ubuntu IRC dataset [22]. Experimental results show that our proposed method outperforms the present state-of-the-art performance by margins of 1.8% variation of information (VI), 4.6% one-to-one overlap (1-1), 9.9% recall, and 8.8% F1 score, achieving a new state-of-the-art performance on dialogue disentanglement. Third, some analysis experiments are conducted to investigate the influence of using different corpora for domain adaptation on the performance of multi-turn response selection. The results show that the more similar to the task dataset this adaptation corpus is, the more improvement it can help to achieve by incorporating domain-specific knowledge into BERT.

In summary, our contributions of this paper are four-fold:

1) We develop several methods on the basis of deep contextualized utterance representations for all four subtasks of Track 2 in DSTC 8.

2) The released evaluation results on unseen test sets show that our final submitted methods ranked fourth in subtask 1, third in subtask 2, and first in subtask 3 and subtask 4 respectively.

3) We also demonstrate that our methods proposed for subtask 1 and subtask 4 outperform existing ones, and achieve new state-of-the-art performances on the public benchmark datasets for multi-turn response selection and dialogue disentanglement.

4) Detailed ablation studies on our proposed methods are conducted to verify their effectiveness on enhancing model performance.

II. TASK DESCRIPTION

The Track 2 in DSTC 8 focuses on task-oriented multi-turn dialogues. It is divided into four different subtasks. Two datasets are provided, i.e., Ubuntu and Advising, which will be introduced in detail in the experiment section. The series of subtasks have similar structures, but vary in the output space and available context. Detailed descriptions of each subtask are introduced as follows.

Subtask 1 is the basic and conventional task of multi-turn response selection, which aims to select an appropriate response from a list of 100 candidates. It is likely that none of these candidates is correct, i.e., either 99 or 100 will be incorrect. Table I shows a conversation example in the Ubuntu dataset for subtask 1.

Subtask 2 is designed to explore dialogues with more than two participants. It is also in the form of multi-turn response selection as subtask 1. But the utterances in this subtask are from more than two participants and are entangled with each other, which makes a more complicated conversational history. Thus, designing a disentanglement strategy to select the relevant utterances is an essential step for this subtask.

Subtask 3 aims to predict whether and where a dialogue has solved the problem. It is worth noting that the number of decisions varies in different dialogues. If there is one or more decisions in a dialogue, the decisions of Accept or Reject and their corresponding utterance positions should be returned.
TABLE III
AN EXAMPLE OF AN ENTANGLED CONVERSATION IN THE UBUNTU
DATASET FOR SUBTASK 4. HERE, “-” IN LABELS DESCRIBES A LINK
BETWEEN TWO MESSAGES.

<table>
<thead>
<tr>
<th>Index</th>
<th>Conversation Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>1087</td>
<td>Does anyone here have k3b experience? I can record things</td>
</tr>
<tr>
<td></td>
<td>with cdrrecord but k3b won’t let me (k3b setup and it</td>
</tr>
<tr>
<td>1088</td>
<td>I thought universe was unsupported though.</td>
</tr>
<tr>
<td>1089</td>
<td>It is.</td>
</tr>
<tr>
<td>1090</td>
<td>[user]: k3b won’t work with kernel [unk].</td>
</tr>
<tr>
<td>1091</td>
<td>[user]: [has joined #ubuntu].</td>
</tr>
<tr>
<td>1092</td>
<td>[user]: So go back to 2.6.7?</td>
</tr>
<tr>
<td>1093</td>
<td>[user]: Yep, or upgrade to 2.6.9.</td>
</tr>
<tr>
<td>1094</td>
<td>Hmm progression or regression.</td>
</tr>
<tr>
<td>1095</td>
<td>[user]: Thanks. I’ll see if ubuntu has 2.6.9 packaged.</td>
</tr>
<tr>
<td>1096</td>
<td>It will work with the ubuntu kernel.</td>
</tr>
<tr>
<td>1097</td>
<td>[user]: k3b will?</td>
</tr>
<tr>
<td>1098</td>
<td>No, there’s no 2.6.9 in warty or hoary.</td>
</tr>
</tbody>
</table>

Label

| 1087 - 1090 |
| 1088 - 1089 |
| 1090 - 1092 |
| 1092 - 1093 |
| 1093 - 1095 |
| 1095 - 1098 |

Otherwise, No Decision Yet and a position of -1 are required to be returned. In other words, this subtask focuses on dialogue state tracking at a fine granularity for each utterance, rather than at a coarse granularity for the whole dialogue. Table II shows an example in the Advising dataset for subtask 3.

Subtask 4 also focuses on dialogue disentanglement. Given a chat log which is composed of hundreds of utterances, it is designed to identify several sections of conversations from this log, each of which is a coherent section composed of several utterances talking about the same topic. Specifically, every log contains more than 1000 messages including directed messages posted by users and information messages. It contains more than 500 links, and each link indicates that the two linked messages are in the same conversation section in time order. Table III shows an example of an entangled conversation log in the Ubuntu dataset for subtask 4.

III. RELATED WORK

The first two subtasks of Track 2 explore the retrieval-based approach to building dialogue systems, which learns a matching model for a pair of a conversational context and a response candidate. This approach has the advantage of providing informative and fluent responses because it selects a proper response for the current conversation from a repository by means of response selection algorithms [2], [5]–[16]. Early studies on retrieval-based chatbots focused on single-turn response selection [23], [24]. Recently, researchers have extended the focus to the multi-turn conversation, which is more practical for real applications. Some methods on multi-turn response selection concatenated the context utterances literally into a single long sequence, and calculated its matching score with a response candidate [5], [6], [25], [26]. Some other methods kept separate utterances and adopted a representation-interaction-aggregation framework for better performance. For example, a multi-view model [27] was proposed, which included an utterance view and a word view. The sequential matching network (SMN) [7] first matched the response with each utterance and then accumulated the matching information by a recurrent neural network. The deep utterance aggregation (DUA) network [8] refined utterances and employed self-matching attention to route the vital information in each utterance. The deep attention matching (DAM) network [9] constructed representations at different granularities with stacked self-attention and cross-attention. The multi-representation fusion network (MRFN) [10] adopted multiple types of representations. The interactive matching network (IMN) [11] performed global and bidirectional interactions between contexts and responses. The utterance-to-utterance interactive matching network (U2U-IMN) [14] treated both contexts and responses as sequences of utterances when calculating the matching degrees between them. The interaction over interaction (IoI) model [13] performed matching by stacking multiple interaction blocks. The multi-hop selector network (MSN) [15] utilized a multi-hop selector to select the relevant utterances as context. The first attempt to employ pre-trained language models for multi-turn response selection was made by concatenating the context utterances and the response literally and sending them into the model for classification [28]. The top results on these two subtasks both used an ensemble of two different models. In detail, the best system on subtask 1 employed an ensemble of BERT and XLNet, which was further augmented by a context retrieval module to utilize the information of the dialogue pool [29]. Meanwhile, the best system on subtask 2 employed an ensemble of ESIM and BERT, which was further combined with a gradient boosting classifier to take the advantage of both models [30]. Different from the studies mentioned above, our proposed methods for subtasks 1 and 2 incorporate speaker change information and domain specific knowledge into the pre-trained language model for multi-turn response selection.

Subtask 3 aims to predict the decision of each utterance, which is closely related to dialogue sentiment analysis [31]–[33] and dialogue action prediction [34]. Each utterance is tagged with a distinct label out of classes. Typical approaches are hierarchical RNN-based models. The utterance-level RNN captures the semantics within a single utterance and the context-level RNN captures the semantics across the whole dialogue. Our method for this subtask tracks the dialogue state at a fine granularity for each utterance and designs a weighted loss function to alleviate the issue of class imbalance.

Subtask 4 studies the dialogue disentanglement problem. Simultaneous conversations occur not only in informal social interactions but also in multi-party-involved chat in our daily life. Aiming to separate intermingled messages into detached conversations, disentanglement is of vital importance for understanding conversations. The research on dialogue disentanglement can date back to the work [35] which conducted a study on voice conversations among 8-10 people with an average of 1.76 conversations active at a time. Recently, the main framework for dialogue disentanglement is a two-step approach. Firstly, a neural network is used to determine the
A relation between two messages. Then a clustering algorithm is adopted to divide messages into different conversations. In the first step, RNNs were used to model adjacent messages [36]. Convolutional neural networks were also adopted to estimate the conversation-level similarity between closely posted messages [37]. A BERT-based masked hierarchical transformer [38] was proposed to calculate the similarity score by using conversation structures. At the second step, the clustering algorithm was usually designed by setting thresholds [37] or grouping two messages with the highest similarity score into the same conversation. The first end-to-end transition-based model for online dialogue disentanglement has been proposed [39]. The recent availability of a large-scale dataset [22] has made it possible to train complex models and to make comprehensive comparisons among different models. In this paper, our method proposed for subtask 4 makes use of the conversational structure information to help the model to learn better contextualized utterance representations explicitly.

IV. METHODOLOGY

A. Subtask 1: Multi-Turn Response Selection

Our model for subtask 1 is built based on BERT. The model details are introduced as follows.

1) Input Representation: To represent a pair of sentence A and sentence B, the original BERT concatenates this pair of sentence with a [SEP] token. For a given token, its input representation of the original BERT is constructed by summing up its corresponding token embeddings, segment embeddings and position embeddings.

In order to distinguish utterances in a context and to model the speaker change in turn as the conversation progresses, we design two strategies to enhance the contextual information and to construct the input sequence for multi-turn response selection as illustrated in Figure 1.

First, additional speaker embeddings are added to token representations to indicate the speaker’s identity for each utterance. For the conversations between two speakers, two speaker embedding vectors are estimated during the training process. Each vector is added to the tokens uttered by each speaker. This can be extended to conversations among more than two speakers by employing a speaker embedding vector for each speaker.

Second, empirical results of previous study [40] show that segmentation tokens played an important role for multi-turn response selection. In this paper, we extend this idea to model turns and utterances. An [EOU] token and an [EOT] token are inserted at the end of an utterance and a turn respectively, which are expected to model the interactions between utterances in a context implicitly.

2) Output Representation: The first token of each concatenated sequence is the [CLS] token, with its embedding $c \in \mathbb{R}^H$ being used as the aggregated representation for a context-response pair. Then, this embedding is sent into a classifier with a sigmoid output layer as follows,

$$s = \text{sigmoid}(w^T c + b),$$

where $w \in \mathbb{R}^H$ and $b \in \mathbb{R}$ are parameters that need to be estimated during the fine-tuning process. Finally, the classifier returns a score $s$ to denote the matching degree of a context-response pair.

3) Multi-Task Learning for Domain Adaptation: The original BERT is trained on a large text corpus and learns general language representations. In order to incorporate specific in-domain knowledge, a domain adaptation strategy is designed. In addition to the datasets for the four subtasks, DSTC 8 also provides external files, which contain the source data of both Ubuntu and Advising domains. The provided external data is utilized to pre-train our BERT model for domain adaptation. Here, a multi-task learning is performed which optimizes a combination of two loss functions, i.e., a masked language model (MLM) loss and a next sentence prediction (NSP) loss [18].

First, for the MLM loss, we follow the settings in the original BERT by masking some percentage of the input tokens at random and then predicting only those masked tokens in order to train a deep bidirectional representation. In more detail, each word is replaced with the [MASK] token at 80% of the time, with a random word at 10% of the time, and with the original word at 10% of the time.

Second, for the NSP loss, the sentence A and sentence B are constructed with the same method as that used in the fine-tuning process. The positive responses are true responses that follow the context, and the negative responses are randomly sampled. For the Ubuntu dataset, we used title and question as sentence A, and answer as sentence B.
Advising dataset, we use the name of courses as sentence A, and its description as sentence B. The embedding of the \([CLS]\) token is utilized as the aggregated representation for classification. Specifically, the speaker embeddings introduced in Section IV-A1 can be pre-trained by the task of NSP. Without domain adaptation, the speaker embeddings have to be initialized randomly at the beginning of the fine-tuning process.

4) Dynamic Negative Sampling: When constructing the training set, the positive and negative responses are sampled in a ratio of 1:1. Given a context, its positive response is fixed and different negative responses are selected at different epochs. Such dynamic negative sampling is expected to improve the generalization ability of the trained model.

B. Subtask 2: Response Selection with Multiple Participants

Subtask 2 is similar to subtask 1 but relies on additional disentanglement strategy. When more than two speakers are communicating in a common channel, there are often multiple conversation topics occurring simultaneously. In terms of a specific conversation topic, utterances relevant to it are useful and other utterances could be considered as noise for response selection. It is worth noting that BERT is not good at dealing with the out-of-vocabulary issue.

First, we define the speaker who is uttering an utterance as the \(\text{spoken-from}\) speaker, and define the speaker who is receiving an utterance as the \(\text{spoken-to}\) speaker. Each utterance in the datasets usually has the labels of both \(\text{spoken-from}\) and \(\text{spoken-to}\) speakers, which can be extracted from the utterance itself. But some utterances may have only the \(\text{spoken-from}\) speaker label. The \(\text{spoken-to}\) speaker is set to \textit{None} in our experiments when it is unknown. Second, given the \(\text{spoken-from}\) speaker of the response, we select the utterances which have the same \(\text{spoken-from}\) or \(\text{spoken-to}\) speaker as the \(\text{spoken-from}\) speaker of the response. Third, these selected utterances are then organized in their original chronological order and used to form the filtered context. Finally, the utterances selected according to their \(\text{spoken-from}\) or \(\text{spoken-to}\) speaker labels are assigned with two speaker embedding vectors respectively.

C. Subtask 3: Dialogue Success

Subtask 3 is different from the first two subtasks, which aims to predict whether and where a dialogue has solved the problem of course advising. We formulate this problem as a combination of sequence labeling and natural language inference (NLI). Here, each utterance is considered as a unit when performing sequence labeling. Similar to NLI, a three-class classification among \textit{Accept} (Entailment), \textit{Reject} (Contradiction) and \textit{No Decision Yet} (Neutral) is conducted for each unit. In this subtask, a hierarchical RNN-based model is adopted rather than BERT because the former achieved a better performance in our preliminary experiments. The detailed implementation of the hierarchical RNN-based model is introduced as follows.

1) Word Representation: The setting in IMN [11] is followed, which constructs word representations by combining pre-trained word embeddings that are estimated on the task-specific training set with character-level embeddings, in order to deal with the out-of-vocabulary issue.

Formally, assuming there are \(n_u\) utterances in a context, the embeddings of the \(m\)-th utterance in the context are denoted as \(\mathbf{U}_m = \{\mathbf{u}_{m,i}\}_{i=1}^{l_{um}}\), where \(l_{um}\) is the number of words in \(\mathbf{U}_m\). Each \(\mathbf{u}_{m,i}\in \mathbb{R}^d\) is an embedding vector.

2) Utterance Encoding: Each utterance is encoded by a BiLSTM [41] separately at the utterance-level to capture semantic information within an utterance. The calculation can be written as

\[
\hat{\mathbf{u}}_{m,i} = \text{BiLSTM}(\mathbf{U}_{m,i}, i \in \{1, \ldots, l_{um}\}),
\]

where \(\hat{\mathbf{U}}_m = \{\hat{\mathbf{u}}_{m,i}\}_{i=1}^{l_{um}}\) and \(\hat{\mathbf{u}}_{m,i} \in \mathbb{R}^d\).

A pooling operation with a combination of max pooling and last-hidden-state pooling is performed to obtain utterance embeddings as

\[
\hat{\mathbf{u}}_{m}^{agr} = \left[\mathbf{u}_{m,max}, \mathbf{u}_{m,l_{um}}\right], \quad m \in \{1, \ldots, n_u\},
\]

where \(\hat{\mathbf{u}}_{m}^{agr} \in \mathbb{R}^{2d}\).

3) Context Encoding: In order to derive the deep contextualized utterance representations, another context-level BiLSTM is employed to aggregate the contextual information across the whole dialogue by considering each utterance as a unit and organizing them in their original chronological order. The calculation can be written as

\[
\hat{\mathbf{u}}_m = \text{BiLSTM}(\hat{\mathbf{U}}_{m}^{agr}, m), \quad m \in \{1, \ldots, n_u\},
\]

where \(\hat{\mathbf{U}}_{m}^{agr} = \{\mathbf{u}_{m,i}\}_{i=1}^{n_u}\) and \(\hat{\mathbf{u}}_m \in \mathbb{R}^d\).

4) Loss Function: The outputs \(\hat{\mathbf{u}}_m\) of the context-level BiLSTM at each time step are used as the inputs of a multi-layer perceptron (MLP) classifier for the 3-class classification as follows,

\[
s_m = \text{softmax}(W_2 \cdot \text{ReLU}(W_1 \cdot \hat{\mathbf{u}}_m + b_1) + b_2),
\]

where \(W_1 \in \mathbb{R}^{d_h \times d}, b_1 \in \mathbb{R}^{d_h}, W_2 \in \mathbb{R}^{3 \times d_h}\) and \(b_2 \in \mathbb{R}^3\) are parameters estimated during the training process. \(d_h\) denotes the dimension of the hidden layer in the MLP. \(s_m \in \mathbb{R}^3\) denotes the score distribution over the 3 classes. Then, the cross entropy loss for one utterance is computed as

\[
\mathcal{L}_m = -\sum_{n=1}^{3} y_{m,n} \log(s_{m,n}),
\]

where \(y_{m,n} \in \{0,1\}\) denotes the true label for the \(n\)-th class of the \(m\)-th utterance.

Furthermore, we divide utterances in a context into two sets. One set \(S_1\) is composed of utterances with labels of \textit{Accept} or \textit{Reject}. The other \(S_2\) is composed of utterances with labels of \textit{No Decision Yet}. Two different weights \(w_1\) and \(w_2\) are introduced for these two sets respectively. Finally, the total loss is a weighted summation of utterance losses as

\[
\mathcal{L} = \sum_{i \in S_1} w_1 \mathcal{L}_i + \sum_{j \in S_2} w_2 \mathcal{L}_j.
\]
In our experiments, we set $w_1 > w_2$ because most utterances in a context belong to the class of *No Decision Yet* and we want to enforce the model to pay more attention to the utterances with labels of *Accept* and *Reject*.

5) Data Augmentation: One challenge of this subtask is the lack of training examples because the training set is composed of only 500 examples. Thus, we augment the training set by using the paraphrases of the sentences and of the target responses which are provided by the track organizer.

D. Subtask 4: Dialogue Disentanglement

Subtask 4 is another disentanglement task. Given a chat log, we need to identify each link between a pair of messages that belong to the same conversation section. Here, each conversation section talks about the same topic in the chat.

A BERT-based model is proposed for this subtask as shown in Figure 2. First, domain adaptation is adopted for this subtask to incorporate specific in-domain knowledge. Furthermore, in order to detect whether every two messages belong to the same conversation, we need to calculate the similarity for each pair of messages. To this end, we consider each message in the chat log as a *target message* and build a set of *context messages* for this target message. Here, the context messages refer to the $K - 1$ messages occurring before this target message and the target message itself. It is noticeable that the context messages include the target message itself in case that the target message is not in a conversation with any other messages. The hyper-parameter $K$ is set manually to keep a balance between performance and complexity.

As shown in Figure 2, each input sequence is constructed by concatenating the target message with each of its context messages. These input sequences are encoded by the pre-trained BERT model to obtain the aggregated sequence embeddings represented by the [CLS] token. As the Transformer-based BERT captures only the long-term dependency within a message pair, then, a context-level BiLSTM is employed on top of the output of BERT in order to capture the contextual semantics across different messages. We denote the output of the BiLSTM as $M^c \in \mathbb{R}^{H \times K}$, where $H$ denotes the dimension of BERT output, and denote the last column of $M^c$ as $m^t \in \mathbb{R}^H$, which corresponds to using the target message itself as the context message.

Furthermore, in order to model the high-order interactions between the target message and its context messages, we compute the element-wise differences and products between $M^c$ and $m^t$, and form the final representation $M$ for similarity measurement as

$$M = [M^c; M^t \odot M^c; M^t - M^c], \quad (8)$$

where $M^t \in \mathbb{R}^{H \times K}$ is obtained by duplicating the $m^t$ vector $K$ times.

Inspired by the linear ranking model [22] that scores each message pair using a feature-based model, we further augment the high-order representations with various manually designed features. Specifically, these features consist of various properties such as time, directedness, word overlap, and context. We denote the feature matrix as $F \in \mathbb{R}^{N \times K}$, where $N$ denotes the feature dimension. Readers can refer to [22] for more details of these features.

Then, the similarity scores $p \in \mathbb{R}^K$ between the target message and its context messages are calculated as

$$p = \tanh([M; F]^{\top} \cdot w + b), \quad (9)$$

where $[;]$ denotes the concatenation operation, $w \in \mathbb{R}^{4H + N}$ and $b \in \mathbb{R}^K$ are parameters estimated during the training process. Here, we select the context message obtaining the highest score with the target message, indicating which history message or none of them is in the same conversation with the target message.

Finally, three ensemble strategies are employed to further improve the performance of this model.

- **Model-AVG.** The final ensemble model is initialized by averaging the weights of several single models with identical architectures and different random initializations.
- **Probability-AVG.** Similarly, prediction probabilities for each sample are averaged across different models.
- **Vote-AVG.** We employ several models to make vote predictions. The context message which achieves the most votes is considered as our final prediction in the same conversation with the target message.

V. EXPERIMENTS

A. Datasets

Our methods were tested on all four subtasks of Track 2 in DSTC 8. Two datasets were provided under this track. The Ubuntu dataset consists of multi-party conversations extracted from the Ubuntu IRC channel. The Advising dataset contains two party dialogues that simulate a discussion between a student and an academic advisor. The purpose of the dialogues is to guide the student to pick courses that fit not only their curriculum, but also personal preferences about time, difficulty, areas of interest, etc. The Ubuntu dataset was provided for subtask 1, 2 and 4, while the Advising dataset was provided for subtask 1 and 3. Some statistics of these datasets were shown in Table IV.

For better comparison with existing methods, we also evaluated our methods developed for subtask 1 and subtask 4 on...
TABLE IV
STATISTICS OF THE DATASETS THAT OUR MODELS WERE TESTED ON.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Valid</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSTC 8-Track</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subtask 1 Ubuntu</td>
<td>225,367</td>
<td>4827</td>
<td>5529</td>
</tr>
<tr>
<td>Subtask 1 Advising</td>
<td>100,000</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>Subtask 2 Ubuntu</td>
<td>122,626</td>
<td>9565</td>
<td>9027</td>
</tr>
<tr>
<td>Subtask 3 Advising</td>
<td>500</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>Subtask 4 Ubuntu</td>
<td>153</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Public benchmarks

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Valid</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ubuntu V1</td>
<td>1M</td>
<td>0.5M</td>
<td>0.5M</td>
</tr>
<tr>
<td>Ubuntu V2</td>
<td>1M</td>
<td>195k</td>
<td>189k</td>
</tr>
<tr>
<td>Douban</td>
<td>1M</td>
<td>50k</td>
<td>10k</td>
</tr>
<tr>
<td>E-commerce</td>
<td>1M</td>
<td>10k</td>
<td>10k</td>
</tr>
</tbody>
</table>

public benchmarks. First, our method for multi-turn response selection were evaluated on four large-scale public datasets, i.e., Ubuntu Dialogue Corpus V1 [5], Ubuntu Dialogue Corpus V2 [6], Douban Conversation Corpus [7], and E-commerce Dialogue Corpus [8]. Some statistics of these four public datasets were shown in Table IV. Ubuntu Dialogue Corpus V1 and V2 also contain multi-turn dialogues about Ubuntu system troubleshooting in English. Here, we adopted the version of Ubuntu Dialogue Corpus V1 shared by previous study [42], in which numbers, paths and URLs were replaced by placeholders. Compared with Ubuntu Dialogue Corpus V1, the training, validation and test dialogues in the V2 dataset were generated in different periods without overlap. The Douban Conversation Corpus was crawled from a Chinese social network on open-domain topics. It was constructed in a similar way to the Ubuntu corpus. The Douban Conversation Corpus collected responses via a small inverted-index system, and labels were manually annotated. The Douban Conversation Corpus is different from the other three datasets in that it includes multiple correct candidates for a context in the test set, which leads to low $R_n@k$, e.g., if there are 3 correct responses, the maximum $R_{10}@1$ is 0.33. Hence, MAP and MRR are recommended for reference. The E-commerce Dialogue Corpus collected real-world conversations between customers and customer service staff from the largest e-commerce platform in China. In all of the corpora, the positive responses are true responses from humans, and the negative responses are randomly sampled. Second, we compared our method for dialogue disentanglement with previous methods on the benchmark of Ubuntu IRC dataset [22], i.e., the dataset used in subtask 4.

B. Evaluation Metrics

For subtask 1 and subtask 2, each model was tasked with selecting the $k$ best-matched responses from $n$ available candidates for the given conversation context, and we calculated the recall of the true positive replies among the $k$ selected responses, denoted as $R_n@k$, as the main evaluation metric. In addition to $R_n@k$, we considered the mean reciprocal rank (MRR). Finally, the average of $R_{100}@10$ and MRR was considered as the final metric for official ranking. When testing on the public datasets, we used the same evaluation metrics as those used in previous work [5]–[8]. Some metrics like the mean average precision (MAP) and precision-at-one (P@1), were considered specifically for the Douban corpus, following the settings of previous work.

For subtask 3, the accuracy of whole dialogue prediction was considered as the main metric for official ranking. In addition, precision, recall and F1 score value were also evaluated for reference.

For subtask 4, seven clustering metrics were adopted for evaluation, including variation of information (VI), one-to-one overlap (1-1), precision, recall, F1 score, adjusted rand index (ARI) and adjusted mutual information (AMI).

C. Training Details

For subtask 1, the large version of BERT was employed. The Adam method [43] was employed for optimization. The initial learning rate was set to 1e-5 and was linearly decreased by L2 weight decay. The maximum sequence length of the concatenation of a context-response pair was set to 320. The training batch size was set to 32. The maximum number of training epochs was set to 30. The dropout [44] probability of 0.1 is applied to all layers. The candidate pool may not contain the correct response, so we need to choose a threshold. When the probability of positive labels was smaller than the threshold, we predicted that candidate pool did not contain the correct response. The threshold was selected from the range [0.6, 0.65, ... 0.95] based on the validation set performance and was set to 0.95 finally. We used the validation set to determine the stop condition and to select the best model for testing.

For subtask 2, the base version of BERT was employed because the large version could not provide further improvement. The initial learning rate was set to 2e-5. The maximum sequence length of the concatenation of a context-response pair was set to 512. The training batch size was set to 25. The maximum number of training epochs was set to 8. The threshold to decide whether the candidate pool containing the correct response was set to 0.95.

For subtask 3, the word representations were 300-dimensional GloVe embeddings [45], the 100-dimensional embeddings estimated on the training set using the Word2Vec algorithm [46] and the 150-dimensional character-level embeddings with window sizes of {3, 4, 5}, each consisting of 50 filters. The word embeddings were not updated during training. All hidden states of the LSTM had 200 dimensions. The MLP at the prediction had 256 hidden units with ReLU activation. Dropout with a rate of 0.2 was applied to the word embeddings and all hidden layers. The maximum utterance length, maximum number of utterances in a context were set to 30 and 26 respectively. Zeros were padded if the number of utterances in a context was less than 26. Otherwise, we kept the last 26 utterances in the context. The Adam method was employed for optimization with a batch size of 200. The learning rate was initialized as 0.001 and was exponentially decayed by 0.96 every 5000 steps. The loss weights $w_1$ and $w_2$ were set as 2 and 1 heuristically.

For subtask 4, we used the base version of BERT, because no further improvement could be achieved by using its
TABLE V
THE OFFICIAL RESULTS OF OUR PROPOSED METHODS ON THE UNSEEN TEST SETS FOR THE TRACK 2 OF DSTC 8. NA - NOT APPLICABLE.

<table>
<thead>
<tr>
<th>Subtask</th>
<th>Measure</th>
<th>Ubuntu</th>
<th>Advising</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R10/0/1</td>
<td>0.649</td>
<td>0.224</td>
</tr>
<tr>
<td></td>
<td>R10/0/5</td>
<td>0.904</td>
<td>0.526</td>
</tr>
<tr>
<td></td>
<td>R10/0/10</td>
<td>0.949</td>
<td>0.676</td>
</tr>
<tr>
<td></td>
<td>MRR</td>
<td>0.760</td>
<td>0.374</td>
</tr>
<tr>
<td>2</td>
<td>R10/0/1</td>
<td>0.506</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R10/0/5</td>
<td>0.755</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R10/0/10</td>
<td>0.834</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MRR</td>
<td>0.621</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Accuracy</td>
<td>NA</td>
<td>0.802</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td></td>
<td>0.832</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>0.817</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>VI</td>
<td>0.933</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1-1</td>
<td>0.806</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.443</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.496</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>0.468</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARI</td>
<td>0.752</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AMI</td>
<td>0.865</td>
<td></td>
</tr>
</tbody>
</table>

TABLE VI
RESULTS OF DIFFERENT ENSEMBLE MODELS ON THE VALIDATION SET OF SUBTASK 4.

<table>
<thead>
<tr>
<th></th>
<th>VI</th>
<th>1-1</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>ARI</th>
<th>AMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model-AVG</td>
<td>0.939</td>
<td>0.853</td>
<td>0.472</td>
<td>0.520</td>
<td>0.495</td>
<td>0.811</td>
<td>0.862</td>
</tr>
<tr>
<td>Probability-AVG</td>
<td>0.947</td>
<td>0.865</td>
<td>0.521</td>
<td>0.547</td>
<td>0.534</td>
<td>0.831</td>
<td>0.888</td>
</tr>
<tr>
<td>Vote-AVG</td>
<td>0.941</td>
<td>0.844</td>
<td>0.519</td>
<td>0.552</td>
<td>0.535</td>
<td>0.783</td>
<td>0.877</td>
</tr>
<tr>
<td>Probability + Vote</td>
<td>0.948</td>
<td>0.872</td>
<td>0.534</td>
<td>0.561</td>
<td>0.547</td>
<td>0.838</td>
<td>0.894</td>
</tr>
</tbody>
</table>

large version. The initial learning rate was set to 2e-5. The maximum sequence length were set to 100. The batch size was set to 4. The max number of messages which were considered as the context of the target message was set to 50. We also tried to consider the messages after the target message, but no further improvement was achieved. The hidden size of the BiLSTM module were set to 384 in order to make the concatenated output equal to 768 which was the same as the output dimension of BERT. The dimension of the designed features for message pairs N was 77.

All codes were implemented in the TensorFlow framework [48] and have been published to help replicate our results.1

D. Experimental Results

Table V presents the evaluation results of our methods on the four subtasks. We tuned our single models on the validation set and submitted the final results using ensemble models. For subtask 1, subtask 2 and subtask 3, the ensemble models were built by averaging the outputs of five single models with identical architectures and different random initializations. For subtask 4, three ensemble strategies introduced in Section IV-D were evaluated and different strategies required different numbers of models to achieve the best performance. For Model-AVG, the number was set to 2. For both Probability-AVG and Vote-AVG, the number was set to 8. The ensemble results on the validation set of subtask 4 were summarized in Table VI. It shows that, the Probability-AVG and Vote-AVG strategies could reach better performance compared with Model-AVG. The submitted system adopted a combination of the Probability-AVG and Vote-AVG strategies which achieved the best performance on the validation set. Finally, our results ranked fourth in subtask 1, third in subtask 2, and first in subtask 3 and subtask 4 respectively.

Comparing with the top results on subtask 1 and subtask 2, the best systems outperformed our methods by margins of 10.1% on subtask 1 [29] and 15.1% on subtask 2 [30] respectively in terms of the final metric for official ranking. The reason might be that these top results both employed an ensemble of different models to take the advantages of them. These different models provided complementary information to some extent which helped to further improve the performance. Although the strategy of ensemble was also adopted in our method, we just combined a set of identical models with different initializations. Thus, the improvement achieved by the ensemble strategy was limited in our method.

The evaluation results of our method proposed for subtask 1 and previous ones on four public datasets were summarized in Table VII and Table VIII. All the results except ours are from existing literatures. Due to previous methods did not make use of pre-trained language models, we also built a BERT baseline by simply fine-tuning BERT on the training set. The BERT baseline was implemented by considering the concatenation of all context utterances as sentence A and the response as sentence B. Then sentence A and sentence B were concatenated with a [SEP] token. As we can see that, BERT has already outperformed the present models on most metrics, except R10/0/5 on Ubuntu Dialogue Corpus V1 and R10/0/1 on E-commerce Corpus. Furthermore, our proposed method outperforms the other models on all metrics and datasets, which demonstrates its ability to select the best-matched response and its compatibility across domains (system troubleshooting, social network and e-commerce). These results show that our proposed method has achieved a new state-of-the-art performance on these datasets for multi-turn response selection.

Table IX summarized the performance of our method proposed for subtask 4 and previous methods on the test set of the Ubuntu IRC dataset. As we can see, our proposed method outperforms the other ones on all metrics except the Precision of Feedforward - Intersect [22], achieving a new state-of-the-art performance on this dataset for dialogue disentanglement.

VI. ANALYSIS

A. Ablation Studies

To demonstrate the importance of each component in our proposed models, a group of ablation studies were conducted, and the results were reported on the validation set on each subtask, as shown in Table X, Table XI, Table XII and Table XIII respectively.

From Table X, we can see that both the domain adaptation process and using speaker embeddings contributed to the
performance of our final model. Without domain adaptation, the metric of $R_{100@1}$ drops 1.6%, which shows that the external data given by the DSTC 8 package can be utilized by domain adaptation to improve our model. Furthermore, $R_{100@1}$ continues to drop 0.6% after ablating speaker embeddings, which shows the effectiveness of modeling the speaker change during the conversation.

Similarly, the domain adaptation process and the speaker embeddings also benefited subtask 2 as shown in Table XI. In addition, we ablated the disentanglement strategy and truncated the sequence to the max sequence length of BERT by selecting the head or tail part. The performance drops significantly which shows the effectiveness of our disentanglement strategy. The reason of the sharp performance drop may be that selecting the head or tail part directly without performing disentanglement does not filter out the useless noise in the
multi-party conversation. Meanwhile, the sequence truncation results in a loss of useful information due to the maximum length limit.

Table XII shows that enriching the training data with the help of the paraphrase and the weighted loss function are both effective for achieving high accuracy in subtask 3.

As shown in Table XIII, the removals of manually designed features, the adaptation process and the context-level BiLSTM all result in the performance drops. Especially, the contextual information contributes most to the performance.

### B. Adaptation Corpus

Some further analysis on the effect of domain adaptation corpus to the performance of multi-turn response selection was conducted on Ubuntu Dialogue Corpus V2. Here, three different Ubuntu datasets were employed for domain adaptation, i.e., DSTC 8-Track 2, Ubuntu Dialogue Corpus V1, and Ubuntu Dialogue Corpus V2. Then, the fine-tuning process was all performed on the training set of Ubuntu Dialogue Corpus V2. The results on the test set of Ubuntu Dialogue Corpus V2 were shown in Table XIV.

As we can see, the adaptation process can help to improve the performance of response selection no matter which adaptation corpus was used. Furthermore, adaptation and fine-tuning on the same corpus achieved the best performance. One explanation may be that although pre-trained language models are designed to provide general linguistic knowledge, the in-domain knowledge contained by a domain-specific corpus is still beneficial to boost model performance. Thus, the more similar to the task this adaptation corpus is, the more improvement it can help to achieve.

### C. Speaker-Aware Disentanglement Strategy

In order to further verify the effectiveness of the speaker-aware disentanglement strategy, we also applied it to existing models, such as IMN [11] and BERT [18], for response selection. The original IMN did not employ any disentanglement strategy and we implemented it by selecting the last 70 utterances as the context, which achieved a performance of 32.2% $R_{100}@1$. After employing the strategy, about 25 utterances were selected to form the context, which achieved a performance of 37.5% $R_{100}@1$. The BERT baseline was implemented by considering the concatenation of all context utterances as sentence A and the response as sentence B. Due to sequence length limit, the context utterances were truncated by selecting the tail part. Then sentence A and sentence B were concatenated with a [SEP] token. Similar results can also be observed by employing this strategy to BERT, as shown in Table XV, which again verified the effectiveness of the speaker-aware disentanglement strategy.

### VII. Conclusion

This paper describes our methods that are developed for all subtasks of Track 2 in DSTC 8. We have explored several methods of building deep contextualized utterance representations for response selection and dialogue analysis in multi-turn dialogue systems. These methods are designed in order to keep the intrinsic property for each subtask according to different evaluation dimensions. In the released evaluation results of Track 2 in DSTC 8, our proposed models ranked fourth in subtask 1, third in subtask 2, and first in subtask 3 and subtask 4 respectively. Extended experiments on the public benchmark datasets show that our methods achieve new state-of-the-art performances on multi-turn response selection and dialogue disentanglement. Investigating other contextualized utterance representations for multi-turn dialogue will be a part of our future work.
Jia-Chen Gu received a B.S. degree in communication engineering from Hohai University, Nanjing, China, in 2017. He is currently working towards a Ph.D. degree in signal and information processing at the University of Science and Technology of China, Hefei, China. His research interests include natural language understanding and deep learning.

Tianda Li received the B.S. degree from Nankai University, Tianjin, China, in 2018 and the MASc degree from Queen’s University, Kingston, Canada, in 2020. His research interests include machine learning, deep learning, and natural language processing.

Zhen-Hua Ling (M’10) received a B.E. degree in electronic information engineering and M.S. and Ph.D. degrees in signal and information processing from the University of Science and Technology of China, Hefei, China, in 2002, 2005, and 2008. From October 2007 to March 2008, he was a Marie Curie Fellow with the Centre for Speech Technology Research, University of Edinburgh, Edinburgh, U.K. From July 2008 to February 2011, he was a joint Postdoctoral Researcher with the University of Science and Technology of China, and iFLYTEK Company, Ltd., Hefei, China. He is currently an Associate Professor with the University of Science and Technology of China. He also worked at the University of Washington, Seattle, WA, USA, as a Visiting Scholar from August 2012 to August 2013. His research interests include speech processing, speech synthesis, voice conversion, and natural language processing. He was the recipient of the IEEE Signal Processing Society Young Author Best Paper Award in 2010. He is currently an Associate Editor of IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING.

Zhiming Su received a B.E. degree in information engineering from Wuhan University of Technology, Wuhan, China, in 2013, and a M.S. degree in information system from Central China Normal University in 2016. He is now a researcher in iFLYTEK CO., LTD., Hefei, China. His research interests include natural language processing, dialogue system and information retrieval.

Quan Liu received a B.E. degree in electronic information engineering from Hohai University, Nanjing, China, in 2012, and a Ph.D. degree in signal and information processing from the University of Science and Technology of China, Hefei, China, in 2017. He is currently a postdoctoral researcher with University of Science and Technology of China, and iFLYTEK CO., LTD., Hefei, China. His research interests include machine learning for natural language processing.

Yu-Ping Ruan received the B.S. degree from the School of Communication and Information Engineering, Shanghai University, Shanghai, China, in 2015. He received the Ph.D. degree in signal and information processing at the University of Science and Technology of China, Hefei, China, in 2020. His research interests include natural language understanding, natural text generation, and deep learning.

Xiaodan Zhu is an Assistant Professor of the Department of Electrical and Computer Engineering and the Ingenuity Labs Research Institute at Queen’s University, and a Faculty Affiliate at the Vector Institute. He received his Ph.D. from the Department of Computer Science of the University of Toronto in 2010 and Masters of Engineering from the Department of Computer Science and Technology of Tsinghua University in 2000. He was a research scientist at the National Research Council Canada from 2010 to 2017. His research interests are in natural language processing, machine learning, and artificial intelligence. He served as a Co-Chair for the 33rd Canadian Conference on Artificial Intelligence in 2020. He was an ACL-2019 Best Paper Selection Committee Member, COLING-2020 Best Paper Selection Committee Member, Associate Editor for the Computational Intelligence journal, Co-chair for SemEval’20, ‘19, PC Co-Chair for NLPCC’20 and CCKS’19, Senior AC for ACL’21, Workshop Co-chair for COLING’20, Area Chair for various NLP and AI conferences.