Dynamic Entity Memory Network for Dialogue Relational Triplet Extraction

Anonymous ACL submission

Abstract

Relational triplet extraction (RTE) is a crucial task in information extraction and has aroused extensive attention. Although advanced studies on RTE have achieved great progress, they are still insufficient for supporting practical applications, such as dialogue system and information retrieval. In this paper, we focus on relational triplet extraction in dialogue scenarios and introduce a new task named dialogue relational triplet extraction (DRTE). Instead of being treated as static texts like sentences or documents, dialogues should be regarded as dynamic ones generated with the progress of conversations. Thus, it imposes three important challenges, including extracting triplets in real-time with incomplete dialogue context, discovering cross-utterance relational triplets, and perceiving the transition of dialogue topics. To tackle these challenges, we propose a Dynamic Entity Memory Network (DEMN). Specifically, the key of our approach is an attentional context encoder and an entity memory network. The attentional context encoder learns dialogue semantics utterance by utterance and dynamically captures salient contexts for each utterance. The entity memory network is devised to store the entities extracted from previous utterances and for cross-utterance triplets extraction. Meanwhile, it also tracks topic transitions in real-time and forgets the semantics of trivial entities. To verify the effectiveness of our model, we manually build three datasets based on KdConv benchmark. Extensive experimental results demonstrate that our model achieves state-of-the-art performances.

1 Introduction

Relational triplet extraction (RTE) task is an important task in natural language processing field, which aims to identify entities and their relations from unstructure text and organize them in the form of ⟨subject, relation, object⟩. As a crucial task beneficial to many applications such as automatic knowledge base construction and question answering, it has attracted widespread attention. Existing studies deal with this task with different paradigms, including table filling (Miwa and Sasaki, 2014; Bekoulis et al., 2018), sequence labeling (Zheng et al., 2017; Wei et al., 2020; Liu et al., 2020), sequence generation (Zeng et al., 2018b; Sui et al., 2020; Zeng et al., 2020), and so on.

Although these studies have achieved great progress, they generally focus on constructing statics knowledge bases by extracting triplets from sentences or documents such as news and Wikipedia articles, while lacking attention to dialogues. To fill this blank, recent studies (Yu et al., 2020; Chen et al., 2020; Xue et al., 2021) explore dialogue relation extraction task and propose graph-based models to deal with it. However, they mainly extract the relations between pre-defined speakers and speaker-related arguments rather than general knowledges such as ⟨Se7en, Director, David Finch⟩ shown in Figure 1: An example of Dialogue Relational Triplet Extraction (DRTE) task. The entities related to different topics are marked in red.
Figure 1. Besides, these models still treat dialogues as flat long texts and neglect the dynamics of them.

To solve the above problems, we introduce a novel task named dialogue relational triplets extraction (DRTE) task, which aims to dynamically discover general knowledge triplets with the progress of dialogues. The dynamic characteristic of dialogue imposes three pivotal challenges for DRTE task. **First,** utterances of each dialogue are generated in real-time, hence posing a key challenge on how to accurately identify entities and relations with incomplete dialogue, especially when partial components of triplets have not yet appeared. **Second,** utterances are usually short and casual, which leads to plenty of cross-utterance triplets. And some triplets even span more than 10 utterances, such as \langle Se7en, Director, David Fincher \rangle in Figure 1. Therefore, how to properly pair the long-distance entities and predict their relation type is an important challenge. **Third,** dialogues generally have more complex topic transitions, how to adapt to this unique logical structure is a key challenge for discovering triplets.

Facing the aforementioned challenges, we propose a Dynamic Entity Memory Network (DEMN). Specifically, we first devise an attentional context encoder to learn the semantics of dialogue utterance by utterance. When our model receives a real-time utterance, this mechanism first utilizes Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) to capture its local semantics, and then adopts the attention mechanism to learn its contextual semantics. Furthermore, we utilize utterance-level LSTM to track the latent topic transition, and devise an entity memory network with forgetting gate for discovering the long-distance triplets without disturbances from entities unrelated to the current topic. To verify the effectiveness of our model, we make a comprehensive and comparative analysis on three datasets, and the results demonstrate that our model achieves state-of-the-art performances. In summary, our contributions are three-fold:

- We introduce dialogue relational triplet extraction (DRTE) task, which is valuable and crucial for downstream tasks but remains under-investigated.
- We propose a dynamic entity memory network (DEMN). By devising an attentional context encoder and an entity memory network, our model can effectively adapt to the dynamic characteristic of dialogues and accurately extract cross-utterance triplets.
- We manually build three datasets based on Kd-Conv benchmark. Extensive experiments are conducted to verify that our model achieves state-of-the-art performances.

2 Related Work

Extracting relational triplets from unstructure text is an important task in information extraction. Previous researches can be mainly categorized into two types, including relation extraction and joint entity and relation extraction.

Relation extraction task aims to predict the relation between any two pre-defined entities according to the given text. Early studies (Mintz et al., 2009; Zeng et al., 2014) effort on sentence-level relation extraction and propose various approaches to alleviate noisy data from distant supervision, such as multi-instance learning (Riedel et al., 2010; Zeng et al., 2015), reinforcement learning (Feng et al., 2018; Zeng et al., 2018a; Qian et al., 2018b), and adversarial training (Qian et al., 2018a; Wu et al., 2017). Although these approaches can effectively classify relations, they fail to deal with cross-sentence relations which limits their application scenarios. To solve this problem, recent studies focus on document-level relation extraction (Yao et al., 2019) and dialogue relation extraction (Yu et al., 2020), which aim to predict relations via semantics of multiple sentences. And plenty of graph based methods (Nan et al., 2020; Li et al., 2020; Xue et al., 2021; Chen et al., 2020) are proposed to adequately model interactions between entities and the context. But these methods assume that entities are pre-defined, which suffers from error propagation problem in practice.

To solve this problem, some studies (Gupta et al., 2016; Zheng et al., 2017) are dedicated to identify entities and their relations in a joint manner. Considering complex relation structures, a variety of neural networks are proposed to extract overlapped triplets, including sequence-to-sequence models (Zeng et al., 2018b; Nayak and Ng, 2020; Ye et al., 2021), sequence labeling models Liu et al. (2020); Wei et al. (2020), token pair linking model (Wang et al., 2020), and reinforcement learning models (Takanobu et al., 2019; Xiao et al., 2020). However, recent studies generally regard sentences, documents, or dialogues as static text,
which fail to adapt the dynamic characteristic of dialogues. To handle this issues, this paper introduce dialogue relational triplets extraction (DRTE) task which aims to identify entities and their relations in real-time for constructing dynamic knowledge graph. To achieve this goal, we propose a Dynamic Entity Memory Network.

3 Methodology

3.1 Problem Formulation

Given a dialogue \( U = \{u_1, u_2, ..., u_{|U|}\} \) with \(|U|\) utterances, dialogue relational triplet extraction (DRTE) task aims to identify the collection of triplets \( T = \{s_i, r_i, o_i\}_{i=1}^{|T|} \), where \( s_i \), \( o_i \), and \( r_i \) represent the subject, object, and relation type of the \( i\)-th triplet, respectively. To deal with this task, we need to recognize the collection of entities \( E = \{e_1, e_2, ..., e_{|E|}\} \) from the given dialogue and predict the relation \( r \) between any two entities. Note that, each entity is extracted from the dialogue content, and the relation type \( r \) is selected from a pre-defined set \( R = \{R_1, R_2, ..., R_{|R|}\} \).

3.2 Framework

There are three pivotal challenges of DRTE task should be tackle, including learning the dynamic context semantics in real-time, discovering the cross-utterance triplets, and tracking the topic transition. To solve these issues, we propose a dynamic entity memory network (DEMN) mainly consisting of an attentional context encoding layer, an entity memory network, and a triplet extraction layer. The overall framework of DEMN is illustrated in Figure 2. Considering the dynamic nature of dialogues, we perform the utterance encoding, entity recognition, triplet extraction and entity memory utterance by utterance. Concretely, we first devise an attentional context encoding layer to learn the isolated semantics and context semantics of each utterance. Based on the fusion of these two semantics, we utilize a token-pair binary classifier for entity recognition. After that, we adopt a supervised multi-head attention mechanism to discovering the relations between any two entity and obtain the inter-utterance and intra-utterance triplets. Finally, the entities of current utterance are used to update the entity memory network, while the semantics of current utterance is used to track the topic transition and weaken the trivial entities.

3.3 Attentional Context Encoding

To dynamically capture the semantics of the real-time utterances, we divide the encoding layer into three parts, including isolated semantics encoding, context semantics encoding and semantics fusion.

Given the \( t\)-th utterance, we first utilize BERT to encode the isolated semantics without considering the historical dialogue. Formally, we tokenize the utterance with the WordPiece vocabulary (Wu et al., 2016) and obtain the input sequence \( u_t = \{x_{[CLS]}, x_{t,1}, x_{t,2}, ..., x_{|u_t|, t} | x_{[SEP]}\} \), where \([CLS]\), \([SEP]\), and \(|u_t|\) denotes the beginning token, the end token, and the utterance length, respectively. The initial representation \( x_{t,1}\) of each token, which is fed into BERT, is constructed by summing its word embedding, position embedding and segment embedding. We take the output of the last Transformer block in BERT as the isolated semantics \( H_t^s = \{h_{s_{[CLS]}^t}, h_{s_{t,1}}^t, ..., h_{s_{|u_t|, t}}^t | h_{s_{[SEP]}^t}\} \).

Meanwhile, we adopt the scaled dot-product attention mechanism to access the historical information pool and obtain the context semantics. Concretely, given the isolated semantics \( H_t^s \) of the \( t\)-th utterance and the historical information pool \( C_t \) at the \( t\)-th step, the context semantics \( H_t^c \) can be calculated as follows:

\[
H_t^c = \text{softmax} \left( \frac{H_t^s W_s \cdot (C_t W_c)^T}{\sqrt{d_c}} \right) C_t, \quad (1)
\]

where \( W_s \in \mathbb{R}^{d_h \times d_c} \) and \( W_c \in \mathbb{R}^{d_h \times d_c} \) are model parameters, \( d_h \) denotes the dimension of BERT, and \( d_c \) is the middle dimension of the dot-product attention.

Finally, we fuse the isolated semantics and the context semantics as follows:

\[
H_t^f = \tanh \left( H_t^s + H_t^c \right). \quad (2)
\]

The final semantics \( H_t^f \) is used to update the historical information pool and extract triplets. When encoding the first utterance, the history pool is empty and the final semantics \( H_1^f \) is equivalent to the isolated semantics \( H_1^s \). After each utterance encoding, we push the final semantics \( H_t^f \) into the historical information pool to update it:

\[
C_{t+1} = \left[ C_t; H_t^f \right]. \quad (3)
\]

3.4 Entity Memory Network

3.4.1 Memory Updating

Based on the semantics of the given utterances, we first identify the entities existing in them and update the entity memory network with these entities.
To follow the dynamic nature of dialogue and the principle that entities will not cross utterances, we perform entity recognition utterance by utterance. Furthermore, due to the existence of nested entities, such as ‘Denver, Colorado, USA’ and ‘USA’ in Figure 1, we formalize the entity recognition task as a token pair linking task (Wang et al., 2020). Formally, we project the final semantics

$$H_f^{t} = \{h_{[CLS]}^t, h_{1}^t, ..., h_{|u|t}^t, h_{[SEP]}^t\}$$

to two semantic subspaces, corresponding to the start and end of the entity respectively. And the probability that two tokens indicate the boundary of an entity can be calculated via a binary classifier:

$$\alpha_{i,j,t} = \sigma (s_{i,t} \cdot (v_{j,t})^T),$$

where $\sigma (*)$ represents the sigmoid function, $W_s \in \mathbb{R}^{d_h \times d_e}$ and $W_v \in \mathbb{R}^{d_h \times d_e}$ are model parameters, and $d_e$ represents the middle dimension of the token pair linking.

During training, we aim to maximize the likelihood probability of the gold annotations as follows:

$$p(y_{i,j,t}|x_{i,t}, x_{j,t}) = \begin{cases} \alpha_{i,j,t}, & \text{if } y_{i,j,t} = 1 \\ 1 - \alpha_{i,j,t}, & \text{if } y_{i,j,t} = 0 \end{cases},$$

where $y_{i,j,t} = 1$ denotes the fact that the phrase $\{x_{i,t}, ..., x_{j,t}\}$ of the $t$-th utterance is an entity, while $y_{i,j,t} = 0$ denotes the corresponding phrase is not an entity. During testing, the entity $\{x_{i,t}, ..., x_{j,t}\}$ is extracted if $\alpha_{i,j,t}$ is higher than a given entity threshold $\gamma$.

We take the averaged hidden representation between the start and end tokens of each entity as its semantics. And the entity memory is updated via appending the extracted entities of each utterance to it. When the memory slot is full, we discard the entity with the weakest semantics so that new entities can be added.

### 3.4.2 Memory Forgetting

To track the topic transition and weaken the semantics of trivial entities, we devise a memory forgetting mechanism. Since it is difficult to obtain the direct supervision information of the topic transition, we first utilize an utterance-level LSTM to discover the latent core topic. After that, we design a forgetting gate to attenuate the semantics of entities that are not related to the current topic. Formally, at the $t$-th step, the semantics $h_{[CLS]}^{t}$ of the $t$-th utterance is fed into the utterance-level
LSTM, and the hidden representation $a_t$ reflecting the current dialogue topic can be distilled as follows:

$$a_t = \text{LSTM}\left(h^f_{[\text{CLS}],t}, a_{t-1}\right). \quad (8)$$

Afterwards, the trivial semantics of the entity memory can be filtered via the forgetting gate:

$$g_{i,t} = \sigma ([a_t; \mathbf{m}_{i,t}]W_g + b_g), \quad (9)$$

$$\mathbf{m}_{i,t+1} = g_{i,t} \odot \mathbf{m}_{i,t}, \quad (10)$$

where $\mathbf{m}_{i,t}$ denotes the hidden representation of the $i$-th entity memory slot at the $t$-th step, $\odot$ denotes the element-wise multiplication, $W_g \in \mathbb{R}^{d_h \times d_a}$ and $b_g \in \mathbb{R}^{d_h}$ are model parameters. The forgetting gate $g_{i,t} \in [0, 1]^{d_h}$ controls the amount of information flowing from each entity memory slot and updates the entity memory $M_t$ to $M_{t+1}$.

### 3.5 Triplet Extraction

To identify triplets accurately and avoid duplicate entity pairing, we design an inter-utterance triplet extraction module and an intra-utterance triplet extraction module.

Formally, given the $t$-th utterance, we can obtain the collection of entities $E_t = \{e_{1,t}, \ldots, e_{|E_t|,t}\}$ extracted from it and the entity memory $M_t = \{\mathbf{m}_{1,t}, \mathbf{m}_{2,t}, \ldots, \mathbf{m}_{|M_t|,t}\}$ consisting of the historical entities. The intra-utterance triplet extraction module only predicts the relations between any two entities from $E_t$, while the inter-utterance triplet extraction module detects the relations between the entity from $E_t$ and the entity from $M_t$.

For each module, we adopt the supervised multi-head attention mechanism (Liu et al., 2020) to predict the relations. Given two entities $e_i$ and $e_j$, we project them to different relation subspaces and calculate their correlation intensity under each subspace as follows:

$$q^i_l = e_iW^i_q, \quad k^j_l = e_jW^j_k, \quad (11)$$

$$\beta^i_{l,i,j} = \sigma \left( \frac{q^i_l \cdot (k^j_l)^T}{\sqrt{d_r}} \right), \quad (12)$$

where $\beta^i_{l,i,j}$ represents the probability that $(e_i, R_l, e_j)$ is identified as a triplet, and $d_r$ is the dimension of each subspace. The representation $q^i_l$ denotes the semantics of $e_i$ as the subject under the relation $r_l$, while $k^j_l$ is the semantics of $e_j$ as the object under the same relation.

During training, we separately maximize the likelihood probability of the gold inter-utterance triplets and the gold intra-utterance triplets as follows:

$$p_{E \to E} (z_t|E_t) = \prod_{i=1}^{|E_t|} \prod_{j=1}^{|E_t|} \prod_{l=1}^{|R|} p \left( z^i_{l,i,j,t} | e_{i,t}, e_{j,t} \right), \quad (13)$$

$$p_{E \to M} (z_t|M_t, E_t) = \prod_{i=1}^{|E_t|} \prod_{j=1}^{|E_t|} \prod_{l=1}^{|R|} p \left( z^i_{l,i,j,t} | e_{i,t}, m_{j,t} \right), \quad (14)$$

$$p_{M \to E} (z_t|M_t, E_t) = \prod_{i=1}^{|M_t|} \prod_{j=1}^{|E_t|} \prod_{l=1}^{|R|} p \left( z^i_{l,i,j,t} | m_{j,t}, e_{i,t} \right), \quad (15)$$

$$p \left( z^i_{l,i,j,t} | e_{i,t}, e_{j,t} \right) = \begin{cases} \beta^i_{l,i,j,t}, & \text{if } z^i_{l,i,j,t} = 1, \\ 1 - \beta^i_{l,i,j,t}, & \text{if } z^i_{l,i,j,t} = 0 \end{cases}, \quad (16)$$

where $E \to E$ denotes that both the subject and object are from $E_t$. Meanwhile, $E \to M$ denotes that the subject is from $E_t$ and the object is from $M_t$, and the meaning of $M \to E$ is opposite to that. During testing, we extract the triplet if the corresponding $\beta^i_{l,i,j}$ is higher than the given relation threshold $\lambda$.

### 3.6 Joint Learning

To synchronously learn the entity recognition and triplet extraction and make them mutually improve, we combine the binary cross-entropy loss functions of the them to form the entire loss objective of our model:

$$\mathcal{L} (\theta) = \mathcal{L}_E + \mathcal{L}_{E \to E} + \mathcal{L}_{E \to M} + \mathcal{L}_{M \to E}, \quad (17)$$

$$\mathcal{L}_E = - \sum_{t=1}^{|U|} \sum_{i=1}^{|u_t|} \mathcal{L}(y_{i,t} = \eta | x_{i,t}, x_{j,t}), \quad (18)$$

$$\mathcal{L}_{E \to E} = - \sum_{t=1}^{|U|} \sum_{i=1}^{|E_t|} \sum_{j=1}^{|E_t|} p \left( z^i_{l,i,j,t} = \eta | e_{i,t}, e_{j,t} \right), \quad (19)$$

$$\mathcal{L}_{E \to M} = - \sum_{t=1}^{|U|} \sum_{i=1}^{|M_t|} \sum_{j=1}^{|E_t|} p \left( z^i_{l,i,j,t} = \eta | e_{i,t}, m_{j,t} \right), \quad (20)$$
We utilize precision, recall, and F1-score to evaluate the performances on dialogue relational triplet extraction. Specifically, a predicted triplet is correct only if the relation type and the whole spans of two entities are all the same as the golden annotation. For reproducibility, we report the average and standard deviation of testing results over 5 runs with different random seeds. At each run, we select the testing result corresponding to the best performance on development set.

### 4.4 Comparison Methods

To comprehensively analyze the advantages of our model, we compare it with joint methods and pipeline methods. It is worth noting that each comparison method adopts BERT as the encoder for ensuring the fairness.

First, three advanced joint entity and relation extraction models are selected as the comparison methods. We concatenate all the utterances as a long sequence and feed it into these methods.
Table 3: Experimental results.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Film</th>
<th></th>
<th>Music</th>
<th></th>
<th>Travel</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1-score</td>
<td>Precision</td>
<td>Recall</td>
<td>F1-score</td>
</tr>
<tr>
<td>CasRel</td>
<td>71.23 ± 1.41</td>
<td>73.56 ± 0.61</td>
<td>72.36 ± 0.44</td>
<td>74.91 ± 2.29</td>
<td>69.48 ± 1.52</td>
<td>72.06 ± 0.24</td>
</tr>
<tr>
<td>TPLinker</td>
<td>68.25 ± 1.02</td>
<td>70.82 ± 0.29</td>
<td>69.51 ± 0.67</td>
<td>66.89 ± 0.93</td>
<td>69.86 ± 0.87</td>
<td>68.33 ± 0.07</td>
</tr>
<tr>
<td>SPN</td>
<td>69.63 ± 0.51</td>
<td>73.27 ± 1.06</td>
<td>71.40 ± 0.77</td>
<td>77.21 ± 1.18</td>
<td>69.25 ± 0.34</td>
<td>73.01 ± 0.74</td>
</tr>
<tr>
<td>TPBC + ATLOP</td>
<td>71.13 ± 1.11</td>
<td>72.74 ± 1.56</td>
<td>71.91 ± 0.99</td>
<td>70.31 ± 1.54</td>
<td>54.04 ± 0.04</td>
<td>61.10 ± 0.24</td>
</tr>
<tr>
<td>TPBC + SSAN</td>
<td>76.05 ± 0.37</td>
<td>61.12 ± 0.08</td>
<td>67.77 ± 0.18</td>
<td>66.70 ± 2.48</td>
<td>76.23 ± 2.31</td>
<td>71.09 ± 0.41</td>
</tr>
<tr>
<td>DEMN (ours)</td>
<td>73.75 ± 0.24</td>
<td>77.79 ± 0.23</td>
<td>75.72 ± 0.23</td>
<td>74.72 ± 0.65</td>
<td>81.65 ± 0.41</td>
<td>78.03 ± 0.54</td>
</tr>
</tbody>
</table>

Figure 3: Results on ablation study.

- CasRel (Wei et al., 2020) first identifies subjects from text, and then devises multiple relation-specific taggers to extract objects for each subject under each relation type.
- TPLinker (Wang et al., 2020) formulates triplet extraction task as a token pair linking problem. For each possible token pairs, this model utilizes a handshaking tagging scheme to predict whether they indicate the boundary of an entity or the association between subject and object.
- SPN (Sui et al., 2020) transforms triplet extraction task into a set prediction problem and proposed a non-autoregressive decoder with bipartite matching loss function to generate all triplets.

Besides, we also select two document-level relation extraction models for conducting pipeline methods. In the first stage, we utilize the token-pair binary classifier (TPBC) of our model to obtain the collection of entities. In the second stage, we adopt relation extraction models to predict the relation between any two entities. The details are described as follows:
- ATLOP (Zhou et al., 2021) is a document-level relation extraction model. It designs a localized context pooling technique which utilizes the pre-trained attention to discover relevant context for entity pairs.
- SSAN (Xu et al., 2021) utilizes an extension of self-attention mechanism to model co-occurrence and coreference entity structure exhibited in document-level texts.

4.5 Results

The results on dialogue relational triplet extraction are shown in Table 3. According to the results, our model consistently obtains state-of-the-art performances on three datasets. Compared with the best baseline model, our model outperforms CaseRel by 3.36% F1-score on Film dataset and is higher than SPN by 5.01% and 2.64% F1-score on Music and Travel datasets, respectively. Besides, the joint extraction models achieves better performance than pipeline models, which verifies that joint learning can make the entity extraction and relation classification mutual promotion.

Specially, all the comparison methods concatenate utterances into a long text and feed it into BERT for encoding, which can fully exploit the semantics of previous utterances and subsequent utterances for entity extraction and relation detection. However, they does not consider the effect of topic transition between utterances, which leads to mismatch and omission of triplets and harms their performances. Conversely, although our model can only use historical information to encodes semantics utterance by utterance, the entity memory network of our model can effectively track dialogue topics and accurately discover triplets. It is
worth noting that our model can flexibly adapt the
dynamic characteristic of dialogues and process
utterances generated in real-time.

4.6 Ablation Study

To further investigate the origination of the im-
provement of DEMN, we conduct three ablation ex-
periments, including ‘DEMN w/o C’, ‘DEMN w/o
F’, and ‘DEMN w/o ALL’. Specifically, ‘DEMN
w/o C’ does not use context semantics captured by
dot-product attention mechanism, while ‘DEMN
w/o F’ discards utterance-level topic tracking mech-
anism and forgetting gate. And ‘DEMN w/o ALL’
abandons these two parts.

According to Figure 3-5, we can analyze the ab-
lation results from three perspectives. First, we
display the overall performances in Figure 3. Com-
pared with ‘DEMN w/o C’, our model achieves
significant improvements on precision, which ver-
ify the importance of context semantics in reducing
mismatch. Besides, First,

5 Conclusion

In this paper, we introduced a novel task named di-

alogue relational triplet extraction (DRTE) and pro-
posed a dynamic entity memory network (DEMN).
To adapt the dynamic characteristic of dialogue, we
mainly devised an attentional context encoder and
an entity memory network. Specifically, the atten-
tional context encoder learn the semantics of the
given dialogue utterance by utterance, which can
flexibly and efficiently understand the utterances
generated in real time. Furthermore, the entity
memory network with a forgetting gate mechanism
maintains the entities extracted from previous ut-
terances for discovering the long-distance triplets
without disturbances from entities unrelated to the
current topic. To verify the effectiveness of our
model, we constructed three datasets. Extensive
experiments show that our model achieves state-of-
the-art performances.

References

Giannis Bekoulis, Johannes Deleu, Thomas Demeester,
and Chris Develder. 2018. Joint entity recognition
and relation extraction as a multi-head selection prob-

Hui Chen, Pengfei Hong, Wei Han, Navonil Majumder,
and Soujanya Poria. 2020. Dialogue relation extrac-
tion with document-level heterogeneous graph atten-

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and
Kristina Toutanova. 2019. BERT: pre-training of
deep bidirectional transformers for language under-


