Abstract

Although neural models have achieved competitive results in dialogue systems, they have shown limited ability in representing core semantics, such as ignoring important entities. To this end, we exploit Abstract Meaning Representation (AMR) to help dialogue modeling. Compared with the textual input, AMR explicitly provides core semantic knowledge and reduces data sparsity. We develop an algorithm to construct dialogue-level AMR graphs from sentence-level AMRs and explore two ways to incorporate AMRs into dialogue systems. Experimental results on both dialogue understanding and response generation tasks show the superiority of our model. To our knowledge, we are the first to leverage a formal semantic representation into neural dialogue modeling.

1 Introduction

Dialogue systems have received increasing research attention (Wen et al., 2015; Serban et al., 2017; Bao et al., 2020), with much recent work focusing on social chats (Ritter et al., 2011; Li et al., 2017) and task-oriented dialogues (Wen et al., 2017; Dinan et al., 2019). There are two salient subtasks in dialogue modeling, namely dialogue understanding (Choi et al., 2018; Reddy et al., 2019; Yu et al., 2020) and response generation (Li et al., 2017; Budzianowski et al., 2018). The former refers to understanding of semantic and discourse details in a dialogue history, and the latter concerns making a fluent, novel and coherent utterance.

The current state-of-the-art methods employ neural networks and end-to-end training (Sutskever et al., 2014; Bahdanau et al., 2015) for dialogue modeling. For instance, sequence-to-sequence models have been used to encode a dialogue history, before directly synthesizing the next utterance (Vinyals and Le, 2015; Wen et al., 2017; Bao et al., 2020). Despite giving strong empirical results, neural models can suffer from spurious feature associations in their neural semantic representation (Poliak et al., 2018; Kaushik et al., 2020), which can lead to weak robustness, inducing irrelevant dialogue states (Xu and Sarikaya, 2014; Sharma et al., 2019; Rastogi et al., 2019) and generating unfaithful or irrelevant text (Maynez et al., 2020; Niu and Bansal, 2020). As shown in Figure 1, the baseline Transformer model pays attention to the word “lamb” but ignores its surrounding context, which has important contents (marked with squares) that indicate its true meaning, thereby giving an irrelevant response that is related to food. Intuitively, such issues can be alleviated by having a structural representation of semantic information, which treats entities as nodes and builds structural relations between nodes, making it easy to find the most salient context. Explicit structures are also more interpretable compared to
neural representation and have been shown useful for information extraction (Strubell et al., 2018; Sun et al., 2019; Li et al., 2020; Bai et al., 2021; Sachan et al., 2021), summarization (Liu et al., 2015; Hardy and Vlachos, 2018; Liao et al., 2018) and machine translation (Marcheggiani et al., 2018; Song et al., 2019a).

We explore AMR (Banarescu et al., 2013) as a semantic representation for dialogue histories in order to better represent conversations. As shown in the central block of Figure 2, AMR is one type of sentential semantic representations, which models a sentence using a rooted directed acyclic graph, highlighting its main concepts (e.g., “mistake”) and semantic relations (e.g., “ARG0”1), while abstracting away function words. It can thus potentially offer core concepts and explicit structures needed for aggregating the main content in dialogue. In addition, AMR can also be useful for reducing the negative influence of variances in surface forms with the same meaning, which adds to data sparsity.

Existing work on AMR parsing focuses on the sentence level. However, as the left block of Figure 2 shows, the semantic structure of a dialogue history can consist of rich cross-utterance co-reference links (marked with squares) and multiple speaker interactions. To this end, we propose an algorithm to automatically derive dialogue-level AMRs from utterance-level AMRs, by adding cross-utterance links that indicate speakers, identical mentions and co-reference links. One example is shown in the right block of Figure 2, where newly added edges are in color. We consider two main approaches of making use of such dialogue-level AMR structures. For the first method, we merge an AMR with tokens in its corresponding sentence via AMR-to-text alignments, before encoding the resulting structure using a graph Transformer (Zhu et al., 2019). For the second method, we separately encode an AMR and its corresponding sentence, before leveraging both representations via feature fusion (Mangai et al., 2010) or dual attention (Calixto et al., 2017).

We verify the effectiveness of the proposed framework on a dialogue relation extraction task (Yu et al., 2020) and a response generation task (Li et al., 2017). Experimental results show that the proposed framework outperforms previous methods (Vaswani et al., 2017; Bao et al., 2020; Yu et al., 2020), achieving the new state-of-the-art results on both benchmarks. Deep analysis and human evaluation suggest that semantic information introduced by AMR can help our model to better understand long dialogues and improve the coherence of dialogue generation. One more advantage is that AMR is helpful to enhance the robustness and has a potential to improve the interpretability of neural models. To our knowledge, this is the first attempt to leverage the AMR semantic representation into neural networks for dialogue understanding and generation. Our code is available at https://github.com/muyeby/AMR-Dialogue.

2 Constructing Dialogue AMRs

Figure 2 illustrates our method for constructing a dialogue-level AMR graph from multiple utterance-level AMRs. Given a dialogue consisting multiple utterances, we adopt a pretrained AMR parser (Cai and Lam, 2020) to obtain an AMR graph for each utterance. For utterances containing multiple sentences, we parse them into multiple AMR graphs, and mark them belonging to the same utterance. We construct each dialogue AMR graph by making connections between utterance AMRs. In particular, we take three strategies according to speaker, identical concept and co-reference information.

Speaker We add a dummy node and connect it to all root nodes of utterance AMRs. We add speaker tags (e.g., SPEAKER1 and SPEAKER2) to the edges to distinguish different speakers. The dummy node ensures that all utterance AMRs are connected so that information can be exchanged during graph encoding. Besides, it serves as the global root node to represent the whole dialogue.

Identical Concept There can be identical mentions in different utterances (e.g. “possible” in the first and the forth utterances in Figure 2), resulting in repeated concept nodes in utterance AMRs. We connect nodes corresponding to the same non-pronoun concepts by edges labeled with SAME2. This type of connection can further enhance cross-sentence information exchange.

Inter-sentence Co-reference A major challenge for dialogues understanding is posed by pronouns,

1Please refer to PropBank (Kingsbury and Palmer, 2002; Palmer et al., 2005) for more details.
which are frequent in conversations (Grosz et al., 1995; Newman et al., 2008; Quan et al., 2019). We conduct co-reference resolution on dialogue text using an off-to-shelf model\(^3\) in order to identify concept nodes in utterance AMRs that refer to the same entity. For example, in Figure 2, “I” in the first utterance, and “sir” in the second utterance refer to the same entity, SPEAKR1. We add edges labeled with COREF between them, starting from later nodes to earlier nodes (later and earlier here refer to the temporal order of ongoing conversation), to indicate their relation\(^4\).

3 Baseline System

We adopt a standard Transformer (Vaswani et al., 2017) for dialogue history encoding. Typically, a Transformer encoder consists of \(L\) layers, taking a sequence of tokens (i.e., dialogue history) \(S = \{w_1, w_2, ..., w_N\}\), where \(w_i\) is the \(i\)-th token and \(N\) is the sequence length, as input and produces vectorized word representations \(\{h_1^l, h_2^l, ..., h_N^l\}\) iteratively, \(l \in [1, ..., L]\). Overall, a Transformer encoder can be written as:

\[
H = \text{SeqEncoder}(\text{emb}(S)),
\]

where \(H = \{h_1^l, h_2^l, ..., h_N^l\}\), and \(\text{emb}\) denotes a function that maps a sequence of tokens into the corresponding embeddings. Each Transformer layer consists of two sub-layers: a self-attention sub-layer and a position-wise feed forward network. The former calculates a set of attention scores:

\[
\alpha_{ij} = \text{Attn}(h_i, h_j).
\]

\(\text{SeqEncoder}\) is the sequence length, as input and produces

which are used to update the hidden state of \(w_i\):

\[
h_i^l = \sum_{j=1}^{N} \alpha_{ij}(W^V h_j^{l-1}),
\]

where \(W^V\) is a parameter matrix. The position-wise feed-forward (FFN) layer consists of two linear transformations:

\[
\text{FFN}(h) = W_2 \text{ReLU}(W_1 h + b_1) + b_2,
\]

where \(W_1, W_2, b_1, b_2\) are model parameters.

3.1 Dialogue Understanding Task

We take the dialogue relation extraction task (Yu et al., 2020) as an example. Given a dialogue history \(S\) and an argument (or entity) pair \((a_1, a_2)\), the goal is to predict the corresponding relation type \(r \in \mathcal{R}\) between \(a_1\) and \(a_2\).

We follow a previous dialogue relation extraction model (Chen et al., 2020) to feed the hidden states of \(a_1\) and \(a_2\) (denoted as \(h_{a_1}, h_{a_2}\)) into a classifier to obtain the probability of each relation types:

\[
P_{rel} = \text{softmax}(W_3 [h_{a_1}; h_{a_2}] + b_3),
\]

where \(W_3\) and \(b_3\) are model parameters. The \(k\)-th value of \(P_{rel}\) is the conditional probability of \(k\)-th relation in \(\mathcal{R}\).

Given a training instance \((S, a_1, a_2, r)\), the local loss is:

\[
\ell = -\log P(r|S, a_1, a_2; \theta),
\]

where \(\theta\) denotes the set of model parameters. In practice, we use BERT (Devlin et al., 2019) for calculating \(h_{a_1}\) and \(h_{a_2}\), which can be regarded as pre-trained initialization of the Transformer encoder.
3.2 Dialogue Response Generation Task

Given a dialogue history $S$, we use a standard auto-regressive Transformer decoder (Vaswani et al., 2017) to generate a response $Y = \{y_1, y_2, ..., y|_Y\}$. At time step $t$, the previous output word $y_{t-1}$ is firstly transformed into a hidden state $s_t$ by a self-attention layer as Equations 2 and 3. Then an encoder-decoder attention mechanism is applied to obtain a context vector from encoder output hidden states $\{h^L_1, h^L_2, ..., h^L_N\}$:

$$\hat{a}_{ti} = \text{Attn}(s_t, h^L_i),$$

$$c_t = \sum^N_{i=1} \hat{a}_{ti} h^L_i.$$  \hspace{1cm} (7)

The obtained context vector $c_t$ is then used to calculate the output probability distribution for the next word $y_t$ over the target vocabulary:

$$P_{\text{vocab}} = \text{softmax}(W_4 c_t + b_4),$$ \hspace{1cm} (8)

where $W_4, b_4$ are trainable model parameters. The $k$-th value of $P_{\text{vocab}}$ is the conditional probability of the $k$-th word in vocabulary given a dialogue.

Given a dialogue history-response pair $\{S, Y\}$, the model minimizes a cross-entropy loss:

$$\ell = - \sum^{|Y|}_{t=1} log P_{\text{vocab}}(y_t | y_{t-1}, ..., y_1, S; \theta),$$ \hspace{1cm} (9)

where $\theta$ denotes all model parameters.

4 Proposed Model

Our model takes a dialogue history $S$ and the corresponding dialogue AMR as input. Formally, an AMR is a directed acyclic graph $G = (V, E)$, where $V$ denotes a set of nodes (i.e. AMR concepts) and $E$ (i.e. AMR relations) denotes a set of labeled edges. An edge can be further represented by a triple $\langle n_i, r_{ij}, n_j \rangle$, meaning that the edge is from node $n_i$ to node $n_j$ with label $r_{ij}$.

We consider two main ways of making use of dialogue-level AMRs. The first method (Figure 3(a)) uses AMR semantic relations to enrich a textual representation of the dialogue history. We project AMR nodes onto the corresponding tokens, extending Transformer by encoding semantic relations between words. For the second approach, we separately encode an AMR and its sentence, and use either feature fusion (Figure 3(b)) or dual attention (Figure 3(c)) to incorporate their embeddings.

4.1 Graph Encoding

We adopt a Graph Transformer (Zhu et al., 2019) to encode an AMR graph, which extends the standard Transformer (Vaswani et al., 2017) for modeling structural input. A $L$-layer graph Transformer takes a set of node embeddings $\{n_1, n_2, ..., n_M\}$ and a set of edge embeddings $\{r_{ij}|i \in [1, ..., M], j \in [1, ..., M]\}$ as input and produces more abstract node features $\{h^L_1, h^L_2, ..., h^L_M\}$ iteratively, where $l \in [1, ..., L]$. The key difference between a graph Transformer and a standard Transformer is the graph attention layer. Compared with self-attention layer (Equation 2), the graph attention layer explicitly considers graph edges when updating node hidden states. For example, give an edge $\langle n_i, r_{ij}, n_j \rangle$, the attention score $\hat{a}_{ij}$ is calculated

---

5 Similar to the encoder, there is also multi-head attention, a position-wise feed-forward layer and residual connections, which we omit in the equations.

6 If there is no relation between $n_i$ and $n_j$, $r_{ij} = \text{"None"}$. 
We first use the JAMR aligner (Flanigan et al., 2014), to obtain a node-to-word alignment, then adopt the alignment to project the AMR edges onto text with following rules:

\[
\hat{r}_{ij} = \begin{cases} 
  r_{i'j'}, & \text{if } A(n_i) = w_i, A(n_{j'}) = w_j, \\
  \text{Self}, & \text{if } i = j, \\
  \text{None}, & \text{otherwise},
\end{cases}
\]  

(13)

where \( A \) is a one-to-\( K \) alignment (\( K \in \{0, \ldots, N\} \)). In this way, we obtain a projected graph \( \mathcal{G}' = (\mathcal{V}', \mathcal{E}') \), where \( \mathcal{V}' \) represents the set of input words \( \{w_1, w_2, \ldots, w_N\} \) and \( \mathcal{E}' \) denotes a set of word-to-word semantic relations.

Inspired by previous work on AMR graph modeling (Guo et al., 2019; Song et al., 2019b; Sun et al., 2019), we adopt a hierarchical encoder that stacks a sequence encoder and a graph encoder. A sequence encoder (SeqEncoder) transforms a dialogue history into a set of hidden states:

\[
H^S = \text{SeqEncoder}(\text{emb}(S)).
\]  

(14)

A graph encoder incorporates the projected relations features into \( H^S \):

\[
H^G = \text{GraphEncoder}(H^S, \text{emb}(\mathcal{E}')).
\]  

(15)

In addition, we add a residual connection between graph adapter and sequence encoder to fuse word representations before and after refinement (as shown in Figure 3(b)):

\[
H^F = \text{LayerNorm}(H^S + H^G).
\]  

(16)

where \( \text{LayerNorm} \) denotes the layer normalization (Ba et al., 2016). We name the hierarchical encoder as \( \text{Hier} \), which can be used for both dialogue understanding and dialogue response generation.

### 4.3 Leveraging both Text and Structure Cues

We consider integrating both text cues and AMR structure cues for dialogue understanding and response generation, using a dual-encoder network. First, a sequence encoder is used to transform a dialogue history \( S \) into a text memory (denoted as \( H^S = \{h^S_1, h^S_2, \ldots, h^S_N\} \) using Equation 1). Second, the AMR graph \( \mathcal{G} \) is encoded into graph memory (denoted as \( H^G = \{h^G_1, h^G_2, \ldots, h^G_M\} \) by a graph Transformer encoder using Equation 12.

For dialogue understanding (Figure 3(b)) and dialogue response generation (Figure 3(c)), slightly different methods of feature integration are used due to their different nature of outputs.

#### Dialogue Understanding

Similar to Section 4.2, we first use the JAMR aligner to obtain a node-to-word alignment \( \mathcal{A} \). Then we fuse the word and AMR node representations as follows:

\[
\hat{h}_i = \begin{cases} 
  f(h^S_i, h^G_j), & \text{if } \exists j, \mathcal{A}(n_j) = w_i, \\
  f(h^S_i, h^S_0), & \text{otherwise},
\end{cases}
\]  

(17)

where \( h_0 \) is the vector representation of the dummy node (see Figure 2). \( f \) is defined as:

\[
h = \text{LayerNorm}(h_1 + h_2).
\]  

(18)

The fused word representations are then fed into a classifier for relation prediction (Equation 5).

#### Dialogue Response Generation

We replace the standard encoder-decoder attention (Equation 7) with a dual-attention mechanism (Song et al., 2019a). In particular, given a decoder hidden state \( s_t \) at time step \( t \), the dual-attention mechanism calculates a graph context vector \( c^G_t \) and a text context vector \( c^S_t \), simultaneously:

\[
\hat{\alpha}_{ti} = \text{Attn}(s_t, h^S_i), \\
\hat{\alpha}_{tj} = \text{Attn}(s_t, h^G_j), \\
c^S_t = \sum_{i=1}^{N} \hat{\alpha}_{ti} h^S_i, \\
c^G_t = \sum_{j=1}^{M} \hat{\alpha}_{tj} h^G_j.
\]  

(19)
and the final context vector $\hat{c}_t$ is calculated as:

$$c_t = W^c[c^S_t; c^G_t] + b^c,$$

where $W^c$ and $b^c$ are model parameters.

We name the dual-encoder model as Dual.

5 Dialogue Understanding Experiments

We evaluate our model on DialogRE (Yu et al., 2020), which contains totally 1,788 dialogues, 10,168 relational triples and 36 relation types in total. On average, a dialogue in DialogRE contains 4.5 relational triples and 12.9 turns. We report experimental results on both original (v1) and updated (v2) English version.\(^7\)

5.1 Settings

We adopt the same input format and hyperparameter settings as Yu et al. (2020) for the proposed model and baselines. In particular, the input sequence is constructed as $[CLS]d[SEP]a_1[SEP]a_2[SEP]$, where $d$ denotes the dialogue, and $a_1$ and $a_2$ are the two associated arguments. In the BERT model of Yu et al. (2020), only the hidden state of the $[CLS]$ token is fed into a classifier for prediction, while our baseline (BERT\(_c\)) additionally takes the hidden states of $a_1$ and $a_2$. All hyperparameters are selected by prediction accuracy on validation dataset (See Table 6 for detailed hyperparameters).

5.2 Main Results

Table 1 shows the results of different systems on DialogRE. We compare the proposed model with two BERT-based approaches, BERT and BERT\(_s\). Based on BERT, BERT\(_s\) (Yu et al., 2020) highlights speaker information by replacing speaker arguments with special tokens. For completeness, we also include recent methods, such as AGGCN (Guo et al., 2019), LSR (Nan et al., 2020) and DHGAT (Chen et al., 2020). BERT\(_c\) and Hier, Dual represent our baseline and the proposed models, respectively.

By incorporating speaker information, BERT\(_s\) gives the best performance among the previous system. Our BERT\(_c\) baseline outperforms BERT\(_s\) by a large margin, as BERT\(_c\) additionally considers argument representations for classification. Hier significantly ($p < 0.01$)\(^8\) outperforms BERT\(_c\) in all settings, with 1.4 points of improvement in terms of F1 score on average. A similar trend is observed under F1\(_c\). This shows that semantic information in AMR is beneficial to dialogue relation extraction, since AMR highlights core entities and semantic relations between them. Dual obtains slightly better results than Hier, which shows effect of separately encoding a semantic structure.

Finally, the standard deviation values of both Dual and Hier are lower than the baselines. This indicates that our approaches are more robust regarding model initialization.

5.3 Impact of Argument Distance

We split the dialogues of the DialogRE (v2) devset into five groups by the utterance-based distance between two arguments. As shown in Figure 4, Dual gives better results than BERT\(_c\) except when

---

\(^{7}\)https://dataset.org/dialogre/

\(^{8}\)We use pair-wised $t$-test.
Figure 4: The performance of BERTc (Baseline) and Dual (Ours) regarding argument distances.

the argument distance is less than 5. In particular, Dual surpasses BERTc by a large margin when the arguments distance is greater than 20. The comparison indicates that AMR can help a model to better handle long-term dependencies by improving the entity recall. In addition to utterance distance, we also consider word distance and observe a similar trend (as shown in Appendix 7).

5.4 Case Study

Figure 5 shows a conversation between a manager and an employee who might have taken a leave. The baseline model incorrectly predicts that the relation between two interlocutors is parent and child. It might be influenced by the last sentence in the conversation, assuming that it is a dialogue between family members. However, the proposed model successful predicts the interlocutors’ relation, suggesting it can extract global semantic information in the dialogue from a comprehensive perspective.

6 Response Generation Experiments

We conduct experiments on the DailyDialog benchmark (Li et al., 2017), which contains 13,119 daily multi-turn conversations. On average, the number of turns for each dialogue is 7.9, and each utterance has 14.6 tokens.

6.1 Settings

We take Transformer as a baseline. Our hyperparameters are selected by word prediction accuracy on validation dataset. The detailed hyperparameters are given in Appendix (See Table 6).

**Metric** We set the decoding beam size as 5 and adopt BLEU-1/2/3/4 (Papineni et al., 2002) and Distinct-1/2 (Li et al., 2016) as automatic evaluation metrics. The former measures the n-gram overlap between generated response and the target response while the latter assesses the generation diversity, which is defined as the number of distinct uni- or bi-grams divided by the total amount of generated words. In addition, we also conduct human evaluation. Following Bao et al. (2020), we ask annotators who study linguistics to evaluate model outputs from four aspects, which are fluency, coherence, informativeness and overall performance. The scores are in a scale of \{0, 1, 2\}. The higher, the better.

6.2 Automatic Evaluation Results

Table 2 reports the performances of the previous state-of-the-art methods and proposed models on the DailyDialog testset. For the previous methods, PLATO and PLATO w/o L are both Transformer models pre-trained on large-scale conversational data (8.3 million samples) and finetuned on DailyDialog. For completeness, we also report other systems including Seq2Seq (Vinyals and Le, 2015) and iVAE_MTI (Fang et al., 2019).

| Model            | BLEU-1/2/3/4 | Distinct-1/2 |
|------------------|--------------|--------------|
| Seq2Seq†         | 33.6/26.8/-/-| 3.0/12.8     |
| iVAE_MTI         | 30.9/24.9/-/-| 2.9/25.0     |
| PLATO w/o L†     | 40.5/32.2/-/-| 4.6/24.6     |
| PLATO†           | 39.7/31.1/-/-| 5.3/29.1     |
| Transformer      | 38.3/31.7/29.1/27.8 | 5.8/30.5     |
| Hier             | 41.3/35.4/33.2/32.1 | 6.5/32.3     |
| Dual             | 40.8/35.0/32.7/31.5 | **6.6/33.0** |

Table 2: Performance on DailyDialog. Results marked with † are from Bao et al. (2020). Models marked with ‡ requires external corpus for pretraining.
Among the previous systems, PLATO and PLATO w/o L report the best performances. Our Transformer baseline is highly competitive in terms of BLEU and Distinct scores. Compared with the Transformer baseline, both Dual and Hier show better numbers regarding BLEU and Distinct, and the gains of both models are significant ($p < 0.01$). This indicates that semantic information in AMR graphs is useful for dialogue response generation. In particular, the gains come from better recall of the important entities and their relations in a dialogue history, which can lead to generating a more detailed response.

### 6.3 Human Evaluation Results

We conduct human evaluation on randomly selected 50 dialogues and corresponding generated responses of the baseline and our models. As shown in Table 3, the Transformer baseline gives the lowest scores, while Dual sees the highest scores from all aspects. Our main advantage is on the Coherence, meaning that AMRs are effective on recalling important concepts and relations. As the result, it makes it easier for our models to generate coherent replies. Examples are shown in Figure 8 in Appendix. Comparatively, all systems achieve high scores regarding Fluency, suggesting that this aspect is not the current bottleneck for response generation.

### 7 Analysis

This section contains analysis concerning the effects of graph features, dialogue length and model robustness. We use Dual model for experiments since it gives slightly better results than Hier.

### 7.1 Ablation on AMR graph

Table 4 shows the results of our best performing models on the two datasets regarding different configurations on the dialogue AMR graphs. We report the average F1 score for DialogRE and the BLEU-1/Distinct-1 score for DailyDialog. First, using utterance-level AMR improves the text baseline by 1.2 points and 1.5 points with regard to F1 and BLEU-1 scores, respectively. This indicates that the semantic knowledge in formal AMR is helpful for dialogue modeling.

Second, our manually added relations (in Section 2) also lead to improvements, ranging from 0.5 to 1.0 in BLEU-1 score. The speaker relation is the most important for dialogue relation extraction, a possible reason is that DialogRE dataset mainly focus on person entities. Also, co-reference relations help the most in dialogue response generation. The identical concept relations give least improvements among three relations. Finally, combining all relations to build a Dialog-AMR graph achieves best performance on both datasets.

### 7.2 Impact of Dialogue Length

We group the devset of DialogRE (v2) and DailyDialog into five groups according to the number of utterances in a dialogue. Figure 6 summarizes the performance of the baseline and the proposed model on dialogue understanding (DU) and response generation (RG) tasks. In dialogue understanding, our model gives slightly better F1 scores than the baseline when a dialogue has smaller than 12 utterance. The performance improvement is more significant when modeling a long dialogue. The performance improvement is more significant when modeling a long dialogue. The performance improvement is more significant when modeling a long dialogue. The performance improvement is more significant when modeling a long dialogue. The performance improvement is more significant when modeling a long dialogue. The performance improvement is more significant when modeling a long dialogue. The performance improvement is more significant when modeling a long dialogue. The performance improvement is more significant when modeling a long dialogue. The performance improvement is more significant when modeling a long dialogue. The performance improvement is more significant when modeling a long dialogue. The performance improvement is more significant when modeling a long dialogue. The performance improvement is more significant when modeling a long dialogue.

This confirms our motivation that AMR can help to understand long dialogues. In dialogue response generation, our model consistently outperforms the Transformer baseline by a large margin on
| Model     | Original | Paraphrased |
|-----------|----------|-------------|
| Baseline  | 100      | 94.50       |
| Ours      | 100      | **98.50**   |

Table 5: F1 on original and paraphrased testsets.

dialogues of different lengths, still with more improvements on larger dialogues. Overall, these results are consistent with Table 1 and 2, showing that AMR can provide useful semantic information and alleviate the issue of long-range dependency.

7.3 Robustness Against Input

Recent studies show that neural network-based dialog models lack robustness (Shalyminov and Lee, 2018; Einolghozati et al., 2019). We select 100 instances from the testset of DialogRE (v2) where both baseline and our model gives true prediction, before paraphrasing the source dialogues manually (see appendix B.3 for paraphrasing guidelines.). Results on the paraphrased dataset are given in Table 5. The performance of baseline model drop from 100 to 94.5 on paraphrased dataset. By contrast, the result of our model reaches 98.5, 4 points higher than baseline. This confirms our assumption that AMR can reduce data sparsity, thus improve the robustness of neural models.

8 Related Work

Semantic Parsing for Dialogue Some previous work builds domain-specified semantic schema for task-oriented dialogues. For example, in the PEGASUS (Zue et al., 1994) system, a sentence is first transformed into a semantic frame and then used for travel planning. Wirsching et al. (2012) use semantic features to help a dialogue system perform certain database operations. Gupta et al. (2018) represent task-oriented conversations as semantic trees where intents and slots are tree nodes. They solve intent classification and slot-filling task via semantic parsing. Cheng et al. (2020) design a rooted semantic graph that integrates domains, verbs, operators and slots in order to perform dialogue state tracking. All these structures are designed for specified task only. In contrast, we investigate a general semantic representation for the modeling of everyday conversations.

Constructing AMRs beyond Sentence Level There are a few attempts to construct AMRs beyond the sentence level. Liu et al. (2015) construct document-level AMRs by merging identical concepts of sentence-level AMRs for abstractive summarization, and Liao et al. (2018) further extend this approach to multi-document summarization. O’Gorman et al. (2018) manually annotate co-reference information across sentence AMRs. We focus on creating conversation-level AMRs to facilitate information exchange more effectively for dialogue modeling.

Bonial et al. (2020) adapt AMRs on dialogues by enriching the standard AMR schema with dialogue acts, tense and aspect, and they construct a dataset consisting of 340 dialogue AMRs. However, they propose theoretical changes in the schema for annotating AMRs, while we explore empirical solutions that leverage existing AMRs of the standard schema on dialogues.

AMR Parsing and Encoding Our work is also related to AMR parsing (Flanigan et al., 2014; Konstas et al., 2017a; Lyu and Titov, 2018; Guo and Lu, 2018; Zhang et al., 2019; Cai and Lam, 2020) and AMR encoding (Konstas et al., 2017b; Song et al., 2018; Zhu et al., 2019; Song et al., 2020; Zhao et al., 2020; Bai et al., 2020). The former task makes it possible to use automatically-generated AMRs for downstream applications, while the latter helps to effectively exploit structural information in AMRs. In this work, we investigate AMRs for dialogue representation and combine AMRs with text for dialogue modeling.

9 Conclusion

We investigated the feasibility of using AMRs for dialogue modeling, describing an algorithm to construct dialogue-level AMRs automatically and exploiting two ways to incorporate AMRs into neural dialogue systems. Experiments on two benchmarks show advantages of using AMR semantic representations model on both dialogue understanding and dialogue response generation.

Acknowledgments

Yue Zhang is the corresponding author. We would like to thank the anonymous reviewers for their insightful comments and Jinhao Jiang for his help for data preparation. This work has been supported by Tencent AI Lab Rhino-Bird Focused Research Program. It also receives support from the Westlake University and Bright Dream Joint Institute for Intelligent Robotics, and a research grant from Rxhui Inc.
References

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. CoRR, abs/1607.06450.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Xuefeng Bai, Pengbo Liu, and Yue Zhang. 2021. Investigating typed syntactic dependencies for targeted sentiment classification using graph attention neural network. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29:503–514.

Xuefeng Bai, Linfeng Song, and Yue Zhang. 2020. Online back-parsing for AMR-to-text generation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1206–1219. Online. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse.

Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. PLATO: Pre-trained dialogue generation model with discrete latent variable. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 85–96. Online. Association for Computational Linguistics.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-AMR: Abstract Meaning Representation for dialogue. In Proceedings of the 12th Language Resources and Evaluation Conference, pages 684–695, Marseille, France. European Language Resources Association.

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Deng Cai and Wai Lam. 2020. AMR parsing via graph-sequence iterative inference. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1290–1301. Online. Association for Computational Linguistics.

Iacer Calixto, Qun Liu, and Nick Campbell. 2017. Doubly-attentive decoder for multi-modal neural machine translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1913–1924, Vancouver, Canada. Association for Computational Linguistics.

Hui Chen, Pengfei Hong, Wei Han, Navonil Majumder, and Soujanya Poria. 2020. Dialogue relation extraction with document-level heterogeneous graph attention networks. CoRR, abs/2009.05092.

Jianpeng Cheng, Devang Agrawal, Héctor Martínez Alonso, Shruti Bhargava, Joris Driesen, Federico Flego, Dain Kaplan, Dimitri Kartsaklis, Lin Li, Dhivya Piraviperumal, Jason D. Williams, Hong Yu, Diarmuid Ó Séaghdha, and Anders Johannsen. 2020. Conversational semantic parsing for dialog state tracking. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8107–8117, Online. Association for Computational Linguistics.

Eunsool Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net.

Arash Einolghozati, Sonal Gupta, Mrinal Mohit, and Rushin Shah. 2019. Improving robustness of task oriented dialog systems. CoRR, abs/1911.05153.

Le Fang, Chunyuan Li, Jianfeng Gao, Wen Dong, and Changyou Chen. 2019. Implicit deep latent variable models for text generation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3946–3956, Hong Kong, China. Association for Computational Linguistics.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the Abstract Meaning.
Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1906–1919, Online. Association for Computational Linguistics.

Matthew L. Newman, Carla J Groom, Lori D Handelman, and James W Pennebaker. 2008. Gender differences in language use: An analysis of 14,000 text samples. Discourse Processes: A Multidisciplinary Journal, 45(3):211–236.

Tong Niu and Mohit Bansal. 2020. Avgout: A simple output-probability measure to eliminate dull responses. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 8560–8567.

Tim O’Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018. AMR beyond the sentence: the multi-sentence AMR corpus. In Proceedings of the 27th International Conference on Computational Linguistics, pages 3693–3702, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Martha Palmer, Paul R. Kingsbury, and Daniel Gildea. 2005. The proposition bank: An annotated corpus of semantic roles. Comput. Linguistics, 31(1):71–106.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Jun Quan, Deyi Xiong, Bonnie Webber, and Changjian Hu. 2019. GECOR: An end-to-end generative ellipsis and co-reference resolution model for task-oriented dialogue. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4547–4557, Hong Kong, China. Association for Computational Linguistics.

Pushpendre Rastogi, Arpit Gupta, Tongfei Chen, and Mathias Lambert. 2019. Scaling multi-domain dialogue state tracking via query reformulation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers), pages 97–105, Minneapolis, Minnesota. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. Transactions of the Association for Computational Linguistics, 7:249–266.

Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 583–593, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Devendra Singh Sachan, Yuhao Zhang, Peng Qi, and William L. Hamilton. 2021. Do syntax trees help pre-trained transformers extract information? In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021, pages 2647–2661. Association for Computational Linguistics.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Lauren Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA, pages 3295–3301. AAAI Press.

Igor Shalyminov and Sungjin Lee. 2018. Improving robustness of neural dialog systems in a data-efficient way with turn dropout. In The Thirty-second Annual Conference on Neural Information Processing Systems (NIPS) 2018, workshop on Conversational AI: “Today’s Practice and Tomorrow’s Potential.

Sanuj Sharma, Prafulla Kumar Choubey, and Ruihong Huang. 2019. Improving dialogue state tracking by discerning the relevant context. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 576–581, Minneapolis, Minnesota. Association for Computational Linguistics.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019a. Semantic neural machine translation using AMR. Transactions of the Association for Computational Linguistics, 7:19–31.

Linfeng Song, Ante Wang, Jinsong Su, Yue Zhang, Kun Xu, Yubin Ge, and Dong Yu. 2020. Structural information preserving for graph-to-text generation.
Linfeng Song, Yue Zhang, Daniel Gildea, Mo Yu, Zhiguo Wang, and Jinsong Su. 2019b. Leveraging dependency forest for neural medical relation extraction. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 208–218, Hong Kong, China. Association for Computational Linguistics.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1616–1626, Melbourne, Australia. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pages 5027–5038. Association for Computational Linguistics.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. Aspect-level sentiment analysis via convolution over dependency tree. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 5679–5688, Hong Kong, China. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. CoRR, abs/1506.05869.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Günther Wirsching, Markus Huber, Christian Kölbl, Robert Lorenz, and Ronald Römer. 2012. Semantic dialogue modeling. In Anna Esposito, Antonietta M. Esposito, Alessandro Vinciarelli, Rüdiger Hoffmann, and Vincent C. Müller, editors, Cognitive behaviourial systems: COST 2102 International Training School, Dresden, Germany, February 21-26, 2011, volume 7403.

P. Xu and R. Sarikaya. 2014. Contextual domain classification in spoken language understanding systems using recurrent neural network. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 136–140.

Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. 2020. Dialogue-based relation extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4927–4940, Online. Association for Computational Linguistics.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. AMR parsing as sequence-to-graph transduction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 80–94, Florence, Italy. Association for Computational Linguistics.

Yanbin Zhao, Lu Chen, Zhi Chen, Ruisheng Cao, Su Zhu, and Kai Yu. 2020. Line graph enhanced AMR-to-text generation with mix-order graph attention networks. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 732–741, Online. Association for Computational Linguistics.

Jie Zhu, Junhui Li, Muhua Zhu, Longhua Qian, Min Zhang, and Guodong Zhou. 2019. Modeling graph structure in transformer for better AMR-to-text generation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 5459–5468, Hong Kong, China. Association for Computational Linguistics.

Victor Zue, Stephanie Seneff, Joseph Polifroni, Michael Phillips, Christine Pao, David Goddeau, James Glass, and Eric Brill. 1994. PEGASUS: A spoken language interface for on-line air travel
planning. In Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994.
A Model parameters

Table 6 lists all model hyperparameters used for experiments. In particular, we share the word vocabulary of encoder and decoder for response generation. We implement our baselines and proposed model based on Pytorch. The preprocessed data and source code will be released at https://github.com/muyeby/AMR-Dialogue.

B More Experimental Results

B.1 Impact of Argument Distance

In addition to utterance distance used in Figure 4, we also consider word-based distance as a metric to measure argument distance. Figure 7 shows F1 scores of baseline and our model on 5 groups of test instances. It can be seen that our model gives better results than baseline system among all distances longer than 30. In particular, our model surpass baseline by 8 points when argument distance is longer than 120.

B.2 Case Study for Dialogue Response Generation

Figure 8 represents a conversation between a hotel service and a guest who wants to book a room, along with its ground-truth response and model-generated responses. We can observe that Transformer’s output is general and not consistent with dialogue history. While proposed models’ outputs can capture the core information “room” from the history, and are more relevant to the topic. Besides, the output given by proposed model is semantically similar to the ground-truth output, but using novel words to response, indicating that the model not only captures the simple dependency between input and output sentences, but also learns deep semantic information of the dialogue history.

B.3 Paraphrasing Guidelines

We ask annotators to paraphrase the dialogues following 3 guidelines:

• do not change the original meaning.
• paraphrase the sentence by using different lexicon and syntax structures.
• paraphrase the dialogue as much as they can.

We also ask a judge to evaluate whether the paraphrased dialogue (sentences) convey the same meaning of the original ones.

| Dialogue History: |
|-------------------|
| … |
| SPEAKER-1: We have new room rates, sir. Will that be acceptable to you? |
| SPEAKER-2: Well, it depends on the price, of course. What is it? |
| SPEAKER-2: It’s $308 a night. |
| SPEAKER-1: I have no problem with that. |
| SPEAKER-2: Great! Would you prefer smoking or nonsmoking? |
| SPEAKER-1: Definitely nonsmoking. I can’t handle that smell. |

| Ground-Truth: |
|----------------|
| Now, is a queen-size bed okay? |

| Transformer: |
|----------------|
| I’m sorry, sir. I’ll be fine. |

| Ours: |
|----------------|
| That’ll be nonsmoking. Now, do you prefer a single queen-size bed? |

Figure 8: Case study for dialogue response generation.
| Setting | DialogRE | DailyDialog |
|---------|----------|-------------|
| Dropout | 0.1      | 0.1         |
| Encoder Layers | 12      | 4           |
| Attention Heads | 12     | 8           |
| Embedding Size | 768     | 512         |
| Hidden Layer size | 768   | 512         |
| Word Vocabulary size | 31k   | 16k         |
| Feed-Forward Layer size | 3072  | 1024        |
| Number of parameters | 110M  | 38M         |
| Dropout | 0.1      | 0.1         |
| Encoder Layers | 2       | 2           |
| Attention Heads | 8      | 8           |
| Hidden Layer size | 512    | 512         |
| Relation Embedding size | 64    | 64          |
| Feed-Forward Layer size | 1024  | 1024        |
| Number of parameters | 4M    | 4M          |
| Dropout | 0.1      | 0.1         |
| Encoder Layers | 3       | 4           |
| Attention Heads | 8      | 8           |
| Hidden Layer Size | 512    | 512         |
| Relation Embedding Size | 64   | 64          |
| Concept Vocabulary Size | 5.2k  | 10k         |
| Feed-Forward Layer Size | 1024  | 1024        |
| Number of parameters | 11M   | 20M         |
| Optimizer | Adam    | Adam        |
| Batch Size | 48      | 20          |
| Learning Rate | 3e-5   | 1e-4        |
| Training Epoch | 30     | 200         |
| Decoder Layers | -      | 4           |
| Training Device | Tesla V100 | Tesla V100 |
| Training Time | 120min | 48h         |

Table 6: Hyperparameters of our models on DialogRE and DailyDialog.