Semantic-based Pre-training for Dialogue Understanding

Xuefeng Bai, Linfeng Song, Yue Zhang

School of Engineering, Westlake University, China
Tencent AI Lab, Bellevue, WA, USA
Institute of Advanced Technology, Westlake Institute for Advanced Study, China

Abstract

Pre-trained language models have made great progress on dialogue tasks. However, these models are typically trained on surface dialogue text, thus are proven to be weak in understanding the main semantic meaning of a dialogue context. We investigate Abstract Meaning Representation (AMR) as explicit semantic knowledge for pre-training models to capture the core semantic information in dialogues during pre-training. In particular, we propose a semantic-based pre-training framework that extends the standard pre-training framework (Devlin et al., 2019) by three tasks for learning 1) core semantic units, 2) semantic relations and 3) the overall semantic representation according to AMR graphs. Experiments on the understanding of both chit-chats and task-oriented dialogues show the superiority of our model. To our knowledge, we are the first to leverage a deep semantic representation for dialogue pre-training.

1 Introduction

Dialogue systems have attracted increasing attention from both academia and industry researches (Chen et al., 2017; Deriu et al., 2021; Gao et al., 2021a). The tasks can be commonly divided into two categories: task-oriented dialogue systems (Wen et al., 2017; Dinan et al., 2019; Mehri et al., 2020) and chit-chat dialogue systems (Ritter et al., 2011; Li et al., 2017; Yu et al., 2020; Cui et al., 2020; Chen et al., 2021, 2022; Song et al., 2022). The former aims to interact in the context of a specific task, while the latter chats with users without task and domain restrictions. Despite differences in goals, a common challenge for both tasks is understanding the semantic information conveyed in a dialogue history.

Recently, semantic representations from pre-trained language models have achieved remarkable success on a spectrum of dialogue tasks (Wen et al., 2015; Zhang et al., 2020; Wu et al., 2020; Gu et al., 2021; Zeng et al., 2021; Zhang and Zhao, 2021; Cui et al., 2021), where knowledge learned in pre-training over large-scale dialogue corpora can be transferred to downstream applications. Current pre-training techniques typically focus on the surface text. However, they do not explicitly consider deep semantic clues beyond text, which leads to some unexpected behavior, such as paying attention to meaningless words (Mudrakarta et al., 2018), and suffering from spurious feature associations (Kaushik et al., 2020) and adversarial attacks (Jia and Liang, 2017).

Incorporating semantic information into dialogue systems has been shown to be helpful for many downstream tasks, such as dialogue intent prediction (Gupta et al., 2018), dialogue state tracking (Cheng et al., 2020), and dialogue relation extraction (Bai et al., 2021). These methods first parse dialogue turns into semantic structures, and then incorporate them as extra features into neural systems. However, they 1) only focus on domain-specific benchmark data, leaving the general potentiality of semantic structures unexploited; 2) require either human annotations or an external parser to obtain semantic structures, raising costs or/and causing error propagation for real applications.

We present SARA, a Semantic-graph-based
pre-trAining fRamework for diAlogues, aiming to endow a pre-trained dialogue model with a stronger ability to infer semantic structures from conversations by using explicit semantic structures for more fine-grained supervisions. In particular, we exploit the abstract meaning representation (AMR; Banerescu et al. 2013), a fine-grained deep structure widely adopted in semantic parsing (Lyu and Titov, 2018; Zhang et al., 2019; Cai and Lam, 2020; Bevilacqua et al., 2021; Bai et al., 2022) and generation (Konstas et al., 2017; Song et al., 2018; Zhu et al., 2019; Bai et al., 2020; Ribeiro et al., 2021). As shown in Figure 1, AMR represents a sentence using a rooted directed graph, highlighting the core semantic units (e.g., “police”, “hum”, “boy”) in a sentence and connecting them with semantic relations (e.g., “:arg0”, “:time”).

We explicitly leverage AMR graphs for pre-training our dialogue model. As shown in Figure 2, SARA consists of three pre-training sub-tasks: 1) semantic-based mask language modeling, which extends the standard mask language modeling task (Devlin et al., 2019) by paying more attention to core semantic units in a dialogue; 2) semantic relation prediction, which aims to learn semantic relations between words; 3) semantic agreement, which optimizes the overall similarity between a dialogue and its corresponding AMR graph. The SARA combines strengths of both powerful contextualized representation of pre-trained models and explicit semantic knowledge, while eliminating the requirement of an external semantic parser in downstream applications.

We choose BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) models as backbone, which are then continual pre-trained on a large-scale conversation dataset using our framework. Experiments show that our semantic-based framework gives better results than current pre-training methods that use much more training data, achieving new state-of-the-art results on both chitchat understanding (dialogue relation extraction) and task-oriented dialogue understanding tasks (DialogoGLUE benchmark). Our method also gives better results than previous semantic-base systems on downstream tasks, without using an external parser. Further analysis suggests that semantic information introduced by AMR can help our model to better understand semantically complex dialogues. To our knowledge, we are the first to leverage deep semantic representation for dialogue pre-training. Our code and the pre-trained models are available at https://github.com/goodbai-nlp/Sem-PLM.

2 Related Work

Pre-training for Dialogue. Inspired by the success of pre-trained language models in the general domain (Peters et al., 2018; Radford and Narasimhan, 2018; Devlin et al., 2019; Lewis et al., 2020), various pre-trained models have been proposed in the domain of dialogue. DiaLoGPT (Zhang et al., 2020) continual pre-trains a GPT-2 (Radford et al., 2019) model directly on Reddit comments data. ConvRT (Henderson et al., 2019) pre-trains a dual Transformer encoder for the response selection task. PLATO (Bao et al., 2020) introduces a latent variable-based model for dialogue response generation pre-training. TOD-BERT (Wu et al., 2020) pre-trains a Transformer encoder on task-oriented dialogue corpus for task-oriented dialogue applications. MPC-BERT (Gu et al., 2021) continues to pre-train a BERT model with self-supervised tasks based on the interactions among utterances and interlocutors. SPIDER (Zhang and Zhao, 2021) introduces a latent variable-based model for dialogue response generation pre-training. TOD-BERT (Wu et al., 2020) pre-trains a Transformer encoder on task-oriented dialogue corpus for task-oriented dialogue applications. MPC-BERT (Gu et al., 2021) continues to pre-train a BERT model with self-supervised tasks based on the interactions among utterances and interlocutors. SPIDER (Zhang and Zhao, 2021) continues to pre-train a BERT model with auxiliary tasks to predict the utterance order and understand the sentence backbone. DialogLM (Zhong et al., 2022) pre-trains a generative Transformer encoder on long conversations with window-based pre-training tasks. Our work is similar in that we also pre-train a model on the dialogue corpora. However, unlike these previous studies, which focus on text level distributions, we additionally enhance the model with semantic structures.

Semantics for dialogue. Semantic knowledge has been used for both social chat and task-oriented dialogues systems. PEGASUS (Zue et al., 1994) transforms a sentence into a semantic frame which is then used for travel planning. Wirsching et al. (2012) design a dialogue system which performs database operations based on semantic features. Gupta et al. (2018) and Aghajanyan et al. (2020) integrate intents and slots into a semantic tree and solve intent classification and slot-filling tasks as semantic parsing. Cheng et al. (2020) represent task-oriented dialogue as a semantic graph to perform dialogue state tracking. A most related work is Bai et al. (2021), who build dialogue-level AMR graphs for both social chat understanding and dialogue response generation. Our work is similar
in showing the effect of semantic knowledge for improving dialogue understanding. However, different from them, we focus on enhancing the language model with semantic knowledge during pre-training, and our model does not require an external AMR parser in downstream applications.

3 Method

Figure 2 illustrates our semantic-based pre-training framework for dialogues. We take a pre-trained Transformer (Vaswani et al., 2017) encoder as the backbone, using AMR as explicit semantic knowledge to continuously pre-train the model on dialogues in a multitask setting. In particular, the following three semantic-aware tasks are designed:

- Semantics-based masking (Section 3.1).
- Semantic relation prediction (Section 3.2).
- Semantic agreement (Section 3.3).

The former two learn semantic knowledge from AMR nodes and AMR edges, respectively. The last task regularizes the overall representation of a dialogue using graph-level semantic features.

We follow Bai et al. (2021) and construct dialogue-level AMR graphs by 2 steps: 1) building utterance-level AMR graphs by independently transforming utterances into AMR using a pre-trained AMR parser. 2) connecting utterance-level AMR graphs with a root node, where edges are labeled with the corresponding speaker.

Formally, denote an input dialogue sequence as $x = [x_1, x_2, ..., x_n]$, where $n$ is the number of tokens in the dialogue. The corresponding AMR is a directed acyclic graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes a set of nodes (i.e., AMR concepts) and $\mathcal{E}$ (i.e., AMR relations) denotes a set of labeled edges. An edge can be further represented by a triple $\langle v_i, r_{ij}, v_j \rangle$, meaning that the edge is from node $v_i$ to $v_j$ with label $r_{ij}$.

3.1 Task 1: Semantics-guided Masking

We first formally present the vanilla mask language modeling (MLM) setup, before introducing the semantic-guided masking strategy.

Vanilla MLM. Given a sequence of tokens $x$, the standard masking strategy (Devlin et al., 2019) selects a set fraction of tokens positions (denoted as $m = [m_1, m_2, ..., m_k]$) for masking independently at random, and use these “selected” tokens $\{x_i|i \in m\}$ as supervisions to train a language model. Formally, denoting the masked text as $\tilde{x}$, vanilla MLM optimizes the following training objective:

$$\ell_{vanilla\_mlm} = - \sum_{i \in m} \log P(x_i|\tilde{x}), \quad (1)$$

where the conditional probability $P(x_i|\tilde{x})$ is generated by an encoder model with a softmax layer.

Semantics-guided Masking. A salient limitation of vanilla MLM is that it treats all tokens equally, thus potentially wasting resources on tokens that provide little signal (e.g., punctuations and stop words). We introduce a semantic-guided masking strategy, encouraging model to give more attention on semantic-aware units, which are expected to have more influence on text understanding. As shown in Figure 2(b), our semantic-guided masking strategy gives a higher masking probability for tokens (e.g. “police”, “could”, “help”) that contain important semantic information. Formally, we define a token as a semantic-aware unit when it is aligned with an AMR node, according to the AMR-to-text alignment $A^2$ (An example is given in Figure 2(a)). Since pre-trained models typically use a vocabulary with sub-word units (Sennrich et al., 2016), for an alignment pair $(v_i, x_j)$, we extend the alignment as $\langle v_i, \{x_j, x_{j+1}, ..., x_{l-1}\} \rangle$, where the AMR node $v_i$ is aligned a set of all tokens $\{x_j, x_{j+1}, ..., x_{l-1}\}$.

1Please refer to Appendix B for dialogue input format.
which are sub-words of word \( w_j \). For example, in Figure 2, the AMR node “housewife” is aligned with sub-tokens “house” and “wife”.

Denoting \( m' = [m'_1, m'_2, ..., m'_k] \) as token indices selected by the proposed semantic-guided masking strategy, the training objective is:

\[
\ell_{sem_{mlm}} = - \sum_{i \in m'} \log P(x_i | \tilde{x}). \tag{2}
\]

We follow ROBERTA (Liu et al., 2019) and use the dynamic masking, where we generate the masking pattern every step instead of performing masking during data preprocessing.

### 3.2 Task 2: Semantic Relation Prediction

The semantic relation prediction task is designed for learning the semantic relations between words. To this end, we project the edges of each input AMR graph onto the corresponding sentence according to their node-to-word alignments (as shown in Figure 2(c)), before training a predictor to generate the projected edges.

**Relation Projection.** Since AMR relations are defined on AMR nodes instead of words in the dialogue text, we use a node-to-word alignment \( A \) to project the AMR edges \( \mathcal{E} \) onto text with following rules:

\[
\hat{r}_{ij} = \begin{cases} \hat{r}_{ij'}, & \text{if } x_i \in A(v_{ij}), x_j \in A(v_{ij'}), \\ \text{None}, & \text{otherwise.} \end{cases}
\tag{3}
\]

The same strategy in Section 3.1 is used to deal with sub-word tokens.

**Relation Prediction.** We first use a Transformer encoder to generate contextualized word hidden states \( h = [h_1, h_2, ..., h_n] \). Based on that, a deep biaffine neural parser (Dozat and Manning, 2017) is used to predict the relations between words. To determine whether a directed edge (or arc) from \( x_i \) to \( x_j \) exists, the biaffine parser first uses two separate MLPs (denoted as \( MLP^H \) and \( MLP^D \)) to obtain two lower-dimensional representation vectors for each position, then calculates scores via a biaffine operation:

\[
r^H_i, r^D_j = MLP^H(h_i), MLP^D(h_j),
\]

\[
\hat{s}^\text{arc}_{ij} = \begin{bmatrix} r^D_j \end{bmatrix}^T \hat{W}^\text{arc} r^H_i,
\tag{4}
\]

\[
P(y^\text{arc}_{ij} | x) = \text{softmax}_j(s^\text{arc}_{ij}),
\]

where \( r^H_i \) is the representation vector of \( x_i \) as a head word, and \( r^D_j \) denotes the vector of \( x_j \) as a dependent word. \( P(y^\text{arc}_{ij} | x) \) is the probability of the arc \((i, j)\), and \( \hat{W}^\text{arc} \) is a parameter matrix. To calculate the probability of assigning a label \( l \) to the arc \((i, j)\), which is denoted as \( P(y^\text{label}_{ij} | x) \), the biaffine parser uses the same scorer as in Equation 4 but with different parameters for MLPs and biaffines.\(^3\)

The training objective of relation prediction is:

\[
\ell_{rel} = - \sum_{(x_i, \hat{r}_{ij}, x_j) \in \mathcal{E}'} \log P(y^\text{arc}_{ij} | x) P(y^\text{label}_{ij} | x),
\tag{5}
\]

where \( \mathcal{E}' \) represents the projected AMR edges.

### 3.3 Task 3: Semantic Agreement

We encourage the model to learn the overall agreement of a dialogue and its corresponding AMR graph. As shown in Figure 2(d), we use an auxiliary network to encode the AMR, and maximize the similarity score between the hidden states of text and AMR. Following previous work (Konstas et al., 2017), we linearize AMR graphs into a sequence (refer to Figure 2(d) for an example) and use a pre-trained encoder to transform AMR into a set of hidden states.\(^4\)

Formally, defining the linearized AMR graph as \( g = [g_1, g_2, ..., g_m] \), the vector representation of text and its corresponding AMR is calculated as:

\[
\begin{align*}
\begin{align*}
\hat{h}^{\text{text}} &= \text{Pooling}(\text{TextEnc}(x)), \\
\hat{h}^{\text{amr}} &= \text{Pooling}(\text{TextEnc}(g)),
\end{align*}
\tag{6}
\end{align*}
\]

where \( \text{TextEnc}(\cdot) \) and \( \text{TextEnc}(\cdot) \) are text encoder and AMR encoder, respectively. They are initialized with the same weights but updated separately during training. \( \text{Pooling}(\cdot) \) is a function that reduces that sequence of vectors into one vector. Following BERT (Devlin et al., 2019), we feed the hidden state of the first input token into a MLP layer to get the “pooled” vector.

We use the cosine similarity as a distance scoring function and adopt the contrastive learning framework (Hadsell et al., 2006; Frosst et al., 2019; Gao et al., 2021b; Luo et al., 2022) to train our model, with the aim to pulling semantically close text-AMR pairs and pushing apart unpaired examples. In particular, for a given text \( x \), the positive example is its corresponding AMR graph \( g \), the negative examples are the AMR graphs of its...

\(^3\)The biaffine parameter for label scoring is a three dimensional tensor.

\(^4\)We also tried a structure-aware encoder but without observing significant improvements.
neighbor dialogues in the corpus. Formally, let \( h_{i}^{\text{amr}} \) and \( h_{i}^{\text{text}} \) denote the representations of the \( i \)th \( \langle \text{text}, \text{AMR} \rangle \) pair in the dataset, the training objective is:

\[
\ell_\text{sim} = -\log \frac{\exp(\text{sim}(h_i^{\text{text}}, h_i^{\text{amr}})/\tau)}{\sum_{j \in \mathcal{N}(i)} \exp(\text{sim}(h_i^{\text{text}}, h_j^{\text{amr}})/\tau)},
\]

where \( \text{sim}(-, -) \) denotes the cosine similarity, \( \mathcal{N}(i) \) collects neighbor index of the \( i \)th example, and \( \tau > 0 \) denotes the temperature hyper-parameter.

### 3.4 Training
Our model is trained by optimizing the total loss of above 3 tasks:

\[
\ell_{\text{total}} = \ell_{\text{scm}, \text{mlm}} + \alpha \ell_{\text{rel}} + \beta \ell_{\text{sim}},
\]

where \( \alpha \) and \( \beta \) are weighting hyper-parameters for \( \ell_{\text{rel}} \) and \( \ell_{\text{sim}} \), respectively. To make the computational requirements feasible, we do not train our model from scratch, but rather continue training a model that has been pre-trained on textual inputs. Our framework is architecture-flexible and can be be applied to different models such as BERT, RoBERTA, and BART.

### 4 Experiments
We evaluate the effectiveness of our semantic pre-training model on 8 dialogue tasks and compare the results with the state-of-the-art pre-trained and semantic-enriched models.

#### 4.1 Dataset
**Pre-training Corpus.** We continual pre-train our model on the Reddit (Henderson et al., 2019) corpus. After sampling and filtering (refer Appendix A), the dataset comprises 5,864,254 dialogue instances, in total 397 million words. We adopt the state-of-the-art AMRBART (Bai et al., 2022) parser\(^5\) to transform the text into AMR graphs. To obtain the AMR-to-text alignment, we use the JAMR aligner\(^6\) released by Flanigan et al. (2014).

\(^5\)https://github.com/muyeby/AMRBART
\(^6\)https://github.com/jflanigan/jamr

#### 4.2 Settings
**Model Configuration.** We take BERT-base and RoBERTa-base as our backbone model. For Pre-training, AdamW (Loshchilov and Hutter, 2019) is used as an optimizer, with an initial learning rate of \( 1 \times 10^{-5} \). We reduce the learning rate according to a linear scheduler. The batch size is 2048, and the maximum input sequence length is 512. For the hyper-parameters, we empirically set \( \alpha = 0.1 \), \( \beta = 1.0 \), \( \tau = 1.0 \) in our experiments. The pre-training of our model is carried out on 8 Nvidia Tesla V100 32G GPU for 5 epochs, taking about 2 days to reach convergence. For fine-tuning, we follow previous works to set hyper-parameters. More details can be found in Appendix C.

**Metrics.** We use macro F1 and macro F1\(_\text{c}\) for dialogue relation extraction (DialogRE), following Yu et al. (2020). For intent prediction (BANKING77, CLINC150, HWU64), we report the accuracy. Macro F1 (Coope et al., 2020) is adopted for slot filling tasks (RESTAURANT8K, DSTC8). For TOP, we use exact-match, which measures how often the model generates the exact reference structure. For MULTIWOZ, we use the joint goal accuracy following Budzianowski et al. (2018).

---

## Table 1: Statistics of datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>DialogRE</th>
<th>BANKING77</th>
<th>HWU64</th>
<th>CLINC150</th>
<th>REST8K</th>
<th>DSTC8</th>
<th>TOP</th>
<th>MULTIWOZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td>5,997</td>
<td>8,622</td>
<td>8,954</td>
<td>15,000</td>
<td>7,244</td>
<td>5,023</td>
<td>31,279</td>
<td>56,774</td>
</tr>
<tr>
<td>dev</td>
<td>1,914</td>
<td>1,540</td>
<td>1,076</td>
<td>3,000</td>
<td>1,000</td>
<td>602</td>
<td>4,462</td>
<td>7,374</td>
</tr>
<tr>
<td>test</td>
<td>1,862</td>
<td>3,080</td>
<td>1,076</td>
<td>4,500</td>
<td>3,731</td>
<td>1,813</td>
<td>9,042</td>
<td>7,372</td>
</tr>
</tbody>
</table>

---

\(^7\)https://github.com/jflanigan/jamr
We adopt the same implementation and hyper-parameters as Henderson et al. (2020) pre-trains a dual Transformer encoder on the full 2015-2019 Reddit data comprising 654M context and entity tokens for classification. For completeness, we also include recent methods which give the state-of-the-art results, such as GDPNet (Xue et al., 2020), TUCORE-GCN (Lee and Choi, 2021), and Hier (Bai et al., 2021). We follow the implementation and hyper-parameters of BERT for all tasks. The model architectures for about tasks is given in Appendix D.

<table>
<thead>
<tr>
<th>Model</th>
<th>data-v1</th>
<th>data-v2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dev</td>
<td>test</td>
</tr>
<tr>
<td></td>
<td>F1(δ)</td>
<td>F1c(δ)</td>
</tr>
<tr>
<td>GDPNet</td>
<td>67.1 (1.0)</td>
<td>61.5 (0.8)</td>
</tr>
<tr>
<td>TUCORE-GCN</td>
<td>66.8 (0.9)</td>
<td>61.5 (1.0)</td>
</tr>
<tr>
<td>BERT</td>
<td>60.6 (1.2)</td>
<td>55.4 (0.9)</td>
</tr>
<tr>
<td>BERTs</td>
<td>63.0 (1.5)</td>
<td>57.3 (1.2)</td>
</tr>
<tr>
<td>BERTc</td>
<td>66.8 (0.9)</td>
<td>60.9 (1.0)</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>68.0 (1.0)</td>
<td>60.3 (1.0)</td>
</tr>
<tr>
<td>SARA-BERT</td>
<td>68.1 (1.0)</td>
<td>62.1 (0.9)</td>
</tr>
<tr>
<td>SARA-RoBERTa</td>
<td>69.3 (0.9)</td>
<td>62.3 (0.8)</td>
</tr>
</tbody>
</table>

Table 2: Performance on DialogRE. We report the average and the standard deviation computed from 5 runs, best results are marked in bold.

4.3 Compared Models

For Dialogue relation extraction, we compare the proposed model with BERT-based models: **BERT** takes a pre-trained BERT as the dialogue encoder and predicts relation labels using the hidden state of the [CLS] token. **BERTs** (Yu et al., 2020) enhances the speaker representation by marking speaker arguments with special tokens. **BERTc** (Bai et al., 2021) concatenates hidden states of the [CLS] token and entity tokens for classification. For completeness, we also include recent methods which give the state-of-the-art results, such as GDPNet (Xue et al., 2020), TUCORE-GCN (Lee and Choi, 2021), TSP (Zhao et al., 2021) and Hier (Bai et al., 2021). We follow the implementation and hyper-parameters of BERT to evaluate our model.

For DialoGLUE, the compared models include: **BERT** (Devlin et al., 2019) pre-trains a Transformer encoder on large-scale monotonic text. **USE** (Yang et al., 2020) pre-trains a dual Transformer encoder model on multilingual corpus using retrieval focused training tasks. **CONVERT** (654M) (Henderson et al., 2020) pre-trains a dual Transformer encoder on the full 2015-2019 Reddit data comprising 654M context and entity tokens for classification. **CONV-BERT** (700M) (Mehri et al., 2020) fine-tunes a BERT model on 700M Reddit conversational data. We adopt the same implementation and hyper-parameters of CONV-BERT (700M) to conduct experiments on DialoGLUE.

To verify the scalability of the proposed method, we also report results based on the RoBERTa model for all tasks. The model architectures for about tasks is given in Appendix D.

4.4 Main Results

**Results on DialogRE.** Table 2 lists the results of different systems on DialogRE. Among BERT-based models (i.e., BERT, BERTs, BERTc), BERTc reports the best results. Compared with BERTc, SARA-BERT gives significantly ($p < 0.001$) better results on both datasets. In particular, SARA-BERT improves BERTc by 1.4 and 2.2 points in terms of F1 score on two test sets, respectively, indicating that our semantic pre-training framework is beneficial for dialogue relation extraction. The main reason can be that SARA improves the model capacity of understanding entities (which are core semantic units) and the semantic relations between them during pre-training stage.

SARA-BERT achieves better F1 scores than the other state-of-the-art methods. In addition, when using RoBERTa as the backbone, SARA gives consistent improvements. In particular, SARA-RoBERTa achieves 68.1 and 67.8 F1 scores on the test set of data-v1 and data-v2, respectively. To our best knowledge, these are the best-reported results based on RoBERTa-base.

**Results on DialoGLUE.** We report the results of different methods on the DialoGLUE benchmark in Table 3. Compared with BERT, SARA-BERT (6M) gives consistently better results on all 7 datasets, with an improvement of 1.1 point in average. In particular, SARA-BERT (6M) outperforms BERT by 2.1 and 3.0 points on HWU64 and MultiWOZ, respectively, showing that our SARA framework can benefit task-oriented dialogue systems.

Compared with the other state-of-the-art systems, SARA-BERT (6M) obtains better results than USE, because SARA-BERT (6M) is pre-trained on large-scale dialogue corpus. In addition, SARA-BERT (6M) gives highly competitive results than
We compare our full system with the following
without the semantics-guided masking, semantic
(6M), SARA-R
Table 3: Performance on DialoGLUE, best results are in bold. REST8K and BANK stands for RESTAURANT8K and BANKING77, respectively.

<table>
<thead>
<tr>
<th>Model</th>
<th>BANK</th>
<th>HWU64</th>
<th>CLINC150</th>
<th>REST8K</th>
<th>DSTC8</th>
<th>TOP</th>
<th>MULTIWOZ</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>USE</td>
<td>92.81</td>
<td>91.25</td>
<td>95.06</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CONVERT (654M)</td>
<td>93.01</td>
<td>91.24</td>
<td>97.16</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>USE+CONVERT (654M)</td>
<td>93.36</td>
<td>92.62</td>
<td>97.16</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CONVBERT (700M)</td>
<td>93.44</td>
<td>92.38</td>
<td>97.11</td>
<td>95.44</td>
<td>91.20</td>
<td>82.08</td>
<td>56.56</td>
<td>86.89</td>
</tr>
<tr>
<td>BERT</td>
<td>93.02</td>
<td>89.87</td>
<td>95.93</td>
<td>95.53</td>
<td>91.05</td>
<td>81.90</td>
<td>56.30</td>
<td>86.08</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>93.16</td>
<td>91.30</td>
<td>96.09</td>
<td>96.27</td>
<td>90.78</td>
<td>81.80</td>
<td>54.95</td>
<td>86.28</td>
</tr>
<tr>
<td>SARA-BERT (6M)</td>
<td>93.47</td>
<td>92.01</td>
<td>96.24</td>
<td>95.92</td>
<td>91.57</td>
<td>82.05</td>
<td>59.33</td>
<td>87.23</td>
</tr>
<tr>
<td>SARA-RoBERTa (6M)</td>
<td>93.64</td>
<td>92.29</td>
<td>96.60</td>
<td>96.74</td>
<td>92.02</td>
<td>82.78</td>
<td>57.52</td>
<td>87.37</td>
</tr>
</tbody>
</table>

Table 4: Validation F1 of DialogRE and DSTC8.

\texttt{CONVERT (654M), USE+CONVERT (654M) and CONVBERT (700M), using significantly fewer data (about 1\% than others). This indicates that our semantic-based pre-training framework is more data-efficient. Finally, similar to SARA-BERT (6M), SARA-RoBERTa (6M) significantly (p < 0.001) outperforms RoBERTa, giving the best results on BANKING77, REST8K, DSTC8 and TOP.}

5 Analysis

5.1 Ablation Study

We compare our full system with the following models: RoBERTa (6M) is continuously pre-trained on the exact same training corpus as our model using corresponding standard pre-training objectives; w/o sem_mlm, w/o rel_pred, and w/o sem_agree denote the models which are trained without the semantics-guided masking, semantic relation prediction, and semantic agreement task, respectively. Table 4 shows the F1 scores on the validation sets of DialogRE and DSTC8. First of all, using dialogue domain data (RoBERTa v.s. RoBERTa (6M)) for pre-training leads to improvements on both tasks. This meets previous observations (Gururangan et al., 2020; Mehr et al., 2020). Also, the semantic-based mask language modeling task (sem_mlm) gives an obvious improvement on DialogRE and a small one on DSTC8. The reason can be that DSTC8 has an average length of 8 tokens, making it easy to understand core semantic units in dialogues. In addition, the performance drops significantly without the relation prediction task (rel_pred), indicating that the rel_pred task is important for dialogue understanding. Furthermore, the semantic agreement task (sem_agree) is helpful for both datasets, showing that the AMR is beneficial to improve the overall semantic representation of dialogue. Finally, by combining dialogue domain data and all pre-training tasks, our final model achieves the best performance on both datasets.

5.2 Effect of Semantic-based Pre-training

To further understand the effectiveness of our semantic-based pre-training framework, we split the test set of DialogRE (v2) into different groups according to semantic complexity and report the performance improvement of SARA-RoBERTa over RoBERTa. In particular, two metrics are considered to measure the semantic complexity of a dialogue: 1) graph size (i.e., the number of nodes in the AMR graph) which records the number of
AMR is a deep semantic structure which consists of the model to understand core semantics, which is utterances. The reason is that SARA encourages bigger when the input dialogue has more than 16 

relations. To study the contribution of such features, we simplify an AMR graph by masking the fine-grained semantic relations, resulting in a graph with frame arguments relations (e.g., :arg0, :arg1, :arg2). We use the simplified graph as explicit semantic knowledge for pre-training and compare it with the standard AMR graph under the same framework.

Table 5 lists the performance of two systems. It can be observed that both simplified graphs and full AMR graphs lead to better performance. Compared with simplified graphs, using full AMR graph for pre-training leads to better results on both DialogRE and DSTC8, showing that the fine-grained semantic features can further improve the model performance.

5.4 Comparison with explicit AMR

Figure 5(a) compares the performance of our model with the method of Bai et al. (2021) which use explicit AMR structures for dialogue applications. We report the F1 score on the test set of DialogRE. Compared with the system of Bai et al. (2021), our model gives comparable results on the validation set, and better results on the test set, without using an external AMR parser. This indicates that 1) our pre-training framework can efficiently transfer the learned semantic information to downstream tasks; 2) large-scale semantic-aware pre-training can give further improvement compared with using

Table 5: F1 on the test set of DialogRE and DSTC8.

<table>
<thead>
<tr>
<th>Model</th>
<th>DialogRE</th>
<th>DSTC8</th>
</tr>
</thead>
<tbody>
<tr>
<td>RoBERTA</td>
<td>65.8</td>
<td>90.78</td>
</tr>
<tr>
<td>SARA-RoBERTA (full)</td>
<td>67.8</td>
<td>92.02</td>
</tr>
<tr>
<td>SARA-RoBERTA (simplified)</td>
<td>67.3</td>
<td>91.34</td>
</tr>
</tbody>
</table>

Figure 4: Test F1 on DialogRE (v2).

Figure 5: (a) Comparison of performance on DialogRE; (b) Comparison of inference speed regarding to dialogue length (measured by number of utterances).

5.3 Impact of AMR Features

AMR is a deep semantic structure which consists of both backbone relations and fine-grained semantic
semantic information in downstream tasks.

As shown in Figure 5(b), our system is significantly faster than the method of Bai et al. (2021) which relies on an external parser. As the dialogue length increases, the performance gap is more obvious. In particular, our system obtains about a 45 times speedup when the input dialogues have an average utterance number of 15.

5.5 Impact of Training Data Scale

Figure 6 shows the model performance regarding different scales of pre-training data. The performance on both DialogRE and DSTC8 datasets increases as the scale of training data grows bigger, with a margin of about 2.0 F1 score on DialogRE. Due to the limitation of computational resources, we do not conduct experiments on larger training corpus and models, and we leave this for a future work.

5.6 Case Study

Figure 7 shows an example conversation from DialogRE dataset. The baseline model (RoBERTa) is misled by sentences last three utterances (marked with underline) where Speaker2 shows an negative emotions towards Rachel Green, and thus incorrectly predicting the relationship between two speakers as negative impression. In contrast, our model (SARA-RoBERTa) predicts the correct relationship, suggesting that our semantic-based pre-training framework helps model to better understand the relationship between entity pairs and avoid focusing on spurious features.

Figure 8 presents a case of dialogue intent prediction. The baseline system pays much attention on word “alarm” while ignores other two core semantic units “minutes” and “bake”, giving an incorrect prediction. Our system successfully predicts the gold intent, because AMR guides our model to discover the core semantic units in the dialogue text.

6 Conclusion

We investigated the abstract meaning representation as explicit semantic clues for dialogue pre-training, using a semantic-based pre-training framework. Experiments on two benchmarks show that the proposed framework is highly effective on both chit-chat understanding and task-oriented dialogue understanding. Our method gives the best results on multiple datasets.

Acknowledgments

Yue Zhang is the corresponding author. We would like to thank anonymous reviewers for their insightful comments. This work is supported by the Zhejiang Province Key Project 2022SDX-HDX0003 and the Tencent AI Lab Rhino-Bird Focused Research Program.

References

Deng Cai and Wai Lam. 2020. AMR parsing via
Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang
Michele Bevilacqua, Rexhina Blloshmi, and Roberto
Siqi Bao, Huang He, Fan Wang, Hua Wu, and
Laura Banarescu, Claire Bonial, Shu Cai, Madalina
Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022.
Graph pre-training for AMR parsing and generation.
In Proceedings of the 60th Annual Meeting of the
Association for Computational Linguistics (Volume
Association for Computational Linguistics.

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 Computa-
tional 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
for Computational Linguistics.

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.

Michele Bevilacqua, Ruxhina Biloshimi, and Roberto
Navigli. 2021. One spring to rule them both:
Symmetric AMR semantic parsing and generation
without a complex pipeline. Proceedings of the AAAI
Conference on Artificial Intelligence, 35(14):12564–
12573.

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang
Tseng, Ihigo Casanueva, Stefan Ultes, Osman
Ramsadan, and Milica Gasić. 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
Association for Computational Linguistics.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz,
Matthew Henderson, and Ivan Vulić. 2020. Efficient
intent detection with dual sentence encoders. In
Proceedings of the 2nd Workshop on Natural
Language Processing for Conversational AI, pages
38–45. Online. Association for Computational
Linguistics.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang
Tang. 2017. A survey on dialogue systems: Recent
advances and new frontiers. SIGKDD Explor.,

Yulong Chen, Yang Liu, Liang Chen, and Yue
dialogue summarization dataset. In Findings of the
Association for Computational Linguistics:
ACL-IJCNLP 2021, pages 5062–5074, Online. Association
for Computational Linguistics.

Yulong Chen, Ming Zhong, Xuefeng Bai, Naihao Deng,
Jing Li, Xianchao Zhu, and Yue Zhang. 2022. The
cross-lingual conversation summarization challenge.
CoRR, abs/2205.00379.

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.

Samuel Coope, Tyler Farghly, Daniela Gerz, Ivan Vulić,
and Matthew Henderson. 2020. Span-ConveRT:
Few-shot span extraction for dialog with pretrained
conversational representations. In Proceedings of
the 58th Annual Meeting of the Association for
Association for Computational Linguistics.

Leyang Cui, Yu Wu, Shujie Liu, and Yue Zhang.
2021. Knowledge enhanced fine-tuning for better
handling unseen entities in dialogue generation.
In Proceedings of the 2021 Conference on Empirical
Methods in Natural Language Processing, pages
2328–2337, Online and Punta Cana, Dominican
Republic. Association for Computational Linguistics.


Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo
Echegoyen, Sophie Rosset, Eneko Agirre, and Mark
Cieliebak. 2021. Survey on evaluation methods for
dialogue systems. Artificial Intelligence Review,


Methods in Natural Language Processing, pages 4269–4282, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.


Tianyang Zhao, Zhao Yan, Yunbo Cao, and Zhoujun Li. 2021. Enhancing dialogue-based relation


<table>
<thead>
<tr>
<th>Param. Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch Size</td>
<td>2048</td>
</tr>
<tr>
<td>Optimizer</td>
<td>AdamW</td>
</tr>
<tr>
<td>Learning Rate (lr)</td>
<td>1e-5</td>
</tr>
<tr>
<td>Lr Scheduler</td>
<td>linear</td>
</tr>
<tr>
<td>Warmup Step</td>
<td>0</td>
</tr>
<tr>
<td>Max Training Epoch</td>
<td>5</td>
</tr>
<tr>
<td>Semantic Masking Prob.</td>
<td>0.2</td>
</tr>
<tr>
<td>Extended Vocabulary Size</td>
<td>30,774</td>
</tr>
<tr>
<td>Max Length (dialogue)</td>
<td>256</td>
</tr>
<tr>
<td>Max Length (AMR)</td>
<td>512</td>
</tr>
<tr>
<td>Mix Precision</td>
<td>fp16</td>
</tr>
<tr>
<td>Parameters (Pre-training)</td>
<td>219M</td>
</tr>
<tr>
<td>Parameters (downstream tasks)</td>
<td>110M</td>
</tr>
<tr>
<td>Training Time</td>
<td>about 45h</td>
</tr>
</tbody>
</table>

Table 6: Hyper-parameters of our models.

C Model Hyper-Parameters

Table 6 lists all model hyper-parameters used for our experiments. The proposed model is implemented based on Pytorch and Huggingface Transformers. Our source code and pre-trained models is released at https://github.com/goodbai-nlp/Sem-PLM.

D Architecture for Downstream Tasks

For all downstream dialogue understanding tasks, we use the pre-trained dialogue model as a dialogue encoder and make prediction based on the encoded hidden states. Taking the BERT-based model as an example, the model architecture of downstream task are:

**Dialogue Relation Extraction:** We concatenate the hidden states of two entities (denoted as $e_1$ and $e_2$) as well as the pooled representation of the [CLS] token into a linear classifier to predict the relation label as:

$$ y = MLP_c([pool(h^{[CLS]}); vec(e_1); vec(e_2)]), $$

(9)

where $MLP_c$ is a linear classifier, and $vec(\cdot)$ selects the encoded representation of the input token. $pool(h^{[CLS]})$ passes the hidden state of the [CLS] token through a linear layer.

**Intent Prediction:** We solve the task as a sequence classification problem, by feeding the pooled hidden state of [CLS] token into a linear classifier to predict the relation label as:

$$ y = MLP_c(pool(h^{[CLS]})). $$

(10)

**Slot Filling:** We represent the problem as IOB tagging, by feeding all hidden state of the input dialogue (denoted by $H$) into a linear classifier and predict the relation label as:

$$ Y = MLP_c(H), $$

(11)

where $H$ denotes the output hidden states, and $Y$ is the output tag sequence.

**Semantic Parsing:** We solve the problem as joint sequence classification and sequence labeling task. Specifically, we predict the intent and slots label as:

$$ y_{intent} = MLP_{intent}(pool(h^{[CLS]})), $$

$$ Y_{slot} = MLP_{slot}(H), $$

(12)

\[8\]https://github.com/huggingface/transformers
where $H$ denotes the output hidden states, and $Y$ is the output tag sequence.

**Dialogue State Tracking:** We follow the TripPy (Heck et al., 2020) framework make prediction, which uses BERT model as encoder and combines BERT with a triple copy strategy to perform state tracking. Please refer the original paper for more details.