

# Multi-Source Probing for Open-Domain Conversational Understanding

Yuanxi Li<sup>1</sup>, Hao Zhou<sup>2</sup>, Jie Zhou<sup>2,3,4</sup>, Minlie Huang<sup>3,4</sup>

<sup>1</sup>University of Illinois at Urbana-Champaign, IL, USA

<sup>2</sup>Pattern Recognition Center, WeChat AI, Tencent Inc., China

<sup>3</sup>The CoAI group, Tsinghua University, Beijing, China

<sup>4</sup>Department of Computer Science and Technology, Tsinghua University, Beijing, China

<sup>1</sup>yuanxi3@illinois.edu, <sup>2</sup>{tuxzhou, withtomzhou}@tencent.com

<sup>3</sup>aihuang@tsinghua.edu.cn

## Abstract

Dialogue comprehension and generation are vital to the success of open-domain dialogue systems. Although pre-trained generative conversation models have made significant progress in generating fluent responses, people have difficulty judging whether they understand and efficiently model the contextual information of the conversation. In this study, we propose a **Multi-Source Probing (MSP)** method to probe the dialogue comprehension abilities of open-domain dialogue models. MSP aggregates features from multiple sources to accomplish diverse task goals and conducts downstream tasks in a generative manner that is consistent with dialogue model pre-training to leverage model capabilities. We conduct probing experiments on seven tasks that require various dialogue comprehension skills, based on the internal representations encoded by dialogue models. Experimental results show that open-domain dialogue models can encode semantic information in the intermediate hidden states, which facilitates dialogue comprehension tasks. Models of different scales and structures possess different conversational understanding capabilities. Our findings encourage a comprehensive evaluation and design of open-domain dialogue models.

## 1 Introduction

Conversational understanding and response generation are critical for the success of open-domain dialogue systems. Recently, pre-trained open-domain dialogue models, including DialoGPT (Zhang et al., 2020), BlenderBot (Roller et al., 2020), and Meena (Adiwardana et al., 2020), have achieved impressive progress in a wide range of conversational tasks. The pre-trained model has become a solid foundation for the downstream fine-tuning process, such as generating empathetic (Zhong et al., 2020) and persona-coherent (Wolf et al., 2019b) responses, delivering knowledge-grounded conversations (Zhao et al., 2020; Wu et al., 2021), and completing task goals (Wu et al., 2020; Peng et al.,

2020). While these generative dialogue models can produce fluent responses, they still have many limitations in conversational understanding (Saleh et al., 2020; Li et al., 2016), leading to irrelevant, repetitive, and generic responses (Li et al., 2017a; Welleck et al., 2019; Cho and Saito, 2021).

Research on conversational understanding can provide a holistic evaluation of dialogue models (Parthasarathi et al., 2020), contributing to the deployment and design of models. However, the analysis of open-domain dialogue models on conversational understanding remains a controversial topic (Dinan et al., 2020; Tao et al., 2018; Ji et al., 2022). Some work (Sankar et al., 2019; Saleh et al., 2020) demonstrates that dialogue models have difficulty in capturing the conversational dynamics in the dialog history and struggle with conversational understanding tasks such as question answering, contradiction inference, and topic determination. In contrast, Parthasarathi et al. (2020) affirms the conversational understanding of open-domain dialogue models and indicates recurrent dialogue models perform better than transformer-based models. These studies have limitations in probing methods and experimental settings (Ravichander et al., 2020), making them not applicable to present large-scale open-domain dialogue models.

In this work, we propose a **Multi-Source Probing (MSP)** method to examine the conversational understanding ability of open-domain dialogue models. Specifically, MSP conducts dialogue comprehension tasks in a generative manner, which is coherent with the pre-trained dialogue generation task to take full advantage of model capabilities. In addition, considering that different tasks require various information of the dialogue context, MSP aggregates features from multiple sources to accomplish diverse tasks. We propose a multi-source cross-attention mechanism to extract local features and adopt a late fusion module to incorporate global features. With the help of these components, MSP

has the capability to evaluate generative dialogue models more accurately and comprehensively.

To expose the validity and reliability of our method, we conduct comprehensive experiments to compare the conventional MLP-based probing approach with MSP. Furthermore, we also set up a series of ablation experiments to verify the necessity of the multiple-source attention mechanism and the late-fusion module.

In order to verify that our method also applies to even more massive scale models, we extend our MSP method to provide insight into larger pre-trained dialogue models. We found that larger models have stronger extraction capability for the information inferred via the pre-trained encoder.

Our study reveals three critical findings:

- Different from the conclusion reached by the vanilla probing method, we find through MSP that encoder hidden states contain more information than original embeddings in pretrained dialogue models, as reflected by the higher accuracy obtained on our probing tasks.
- Generative dialogue models with a single decoder have a worse understanding of the conversation than encoder-decoder-based models, as the uni-directional attention mechanism only encodes partial context (content before each token) information for tokens, leading to asymmetric representations of dialogue history and current utterance.
- Dialogue models can capture the dialogue structure in conversational understanding. Larger dialogue models have a better understanding of conversational information and achieve higher accuracy on probing tasks.

## 2 Related Work

### 2.1 Open-domain Conversational Models

Recently, open-domain conversation systems have been largely advanced due to the increase of dialogue corpus and the development of large-scale pre-training (Devlin et al., 2018; Radford et al., 2019; Brown et al., 2020). Pre-trained open-domain dialogue models, such as DialoGPT (Zhang et al., 2020), BlenderBot (Roller et al., 2020) and Meena (Adiwardana et al., 2020), demonstrate outstanding conversation skills, including empathetic (Zhong et al., 2020) and persona-coherent (Wolf et al., 2019b) response generation, delivering knowledge-grounded conversa-

tions (Zhao et al., 2020; Wu et al., 2021) and completing task goals (Wu et al., 2020; Peng et al., 2020). These dialogue models are capable of fluent response generation, but they still have many conversational understanding limits (Saleh et al., 2020; Li et al., 2016) that result in irrelevant, repetitive, and generic responses (Serban et al., 2017; Li et al., 2017a; Welleck et al., 2019; Cho and Saito, 2021). Several studies (Sankar et al., 2019; Bao et al., 2020) point out that generative dialogue models don't always properly exploit the existing dialog history and they are yet unable to understand the context to provide coherent and engaging conversations. Some work (Komeili et al., 2022; Zhou et al., 2018) introduce external knowledge to enhance conversational understanding, which facilitates the generation of relevant and coherent responses.

### 2.2 Probing Method

With the growing demand for natural language understanding, the probing method has been widely employed in machine translation (Belinkov et al., 2017, 2018; Dalvi et al., 2017; Yawei and Fan, 2021) and knowledge attribution (Alishahi et al., 2017; Beloucif and Biemann, 2021) to assess the linguistic properties of sentence representations learned by models.

Although several studies have been proposed to probe the conversational understanding capability of open-domain dialogue models (Dinan et al., 2020; Tao et al., 2018; Ji et al., 2022), this research area is still controversial. According to certain research (Sankar et al., 2019; Saleh et al., 2020; Das et al., 2020), conversational comprehension tasks including question answering, contradiction inference, and subject determination pose challenges for dialogue models in terms of capturing the conversational dynamics in the dialog history. In contrast, Parthasarathi et al. (2020) validates conversational comprehension of open-domain dialogue models and demonstrates that recurrent dialogue models outperform transformer-based models. In addition, previous work (Saleh et al., 2020; Alt et al., 2020; Ravichander et al., 2020; Parthasarathi et al., 2020; Richardson et al., 2020) usually adopted a certain probing method to perform model-level analysis of dialogue systems, thus they lacked an exhaustive comparison of different probing methods. These studies have reached opposite conclusions for two major reasons. First, these methods usually adopt a shallow Multi-Layer Perceptron (MLP) as the clas-

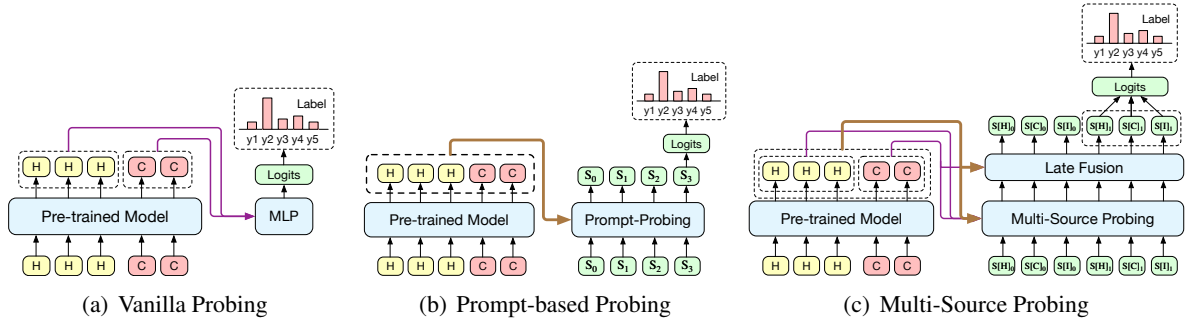


Figure 1: Overview of three probing methods. The MSP method introduces the multi-source attention mechanism to integrate local features from multiple sources and the late fusion module to capture global features, where probing tasks are conducted in a generative manner, which is consistent with the objective of the pre-training task.

sifier, which cannot fully utilize the information encoded in intermediate representations to conduct probing tasks. Second, the experimental settings are also insufficient (Ravichander et al., 2020), including the probing tasks and probed model scales.

### 3 Methodology

#### 3.1 Vanilla Probing Approach

For the vanilla probing method, we first use pre-trained generative dialogue models to extract encoder hidden states and word embeddings corresponding to the probing task texts and then feed the extracted representations to a two-layer Multi-Layer Perceptron (MLP) classifier.

In this way, we calculate the accuracy of probing tasks, which is empirically assumed to reflect the ability of the corresponding model to capture information that is beneficial for the goal of dialogue understanding. Here only the parameters of the classifier are trainable during the training of probing tasks, with encoder parameters kept fixed. The vanilla probing method is shown in Fig 1(a).

First, we extract inner states from the encoder to get the word-level representations on **word embeddings** and **encoder states** in the entire probing task input  $x = [u_1, \dots, u_n]$ , with  $u_n = [w_1^{(n)}, \dots, w_{c_n}^{(n)}]$  denoting the tokens in the  $n_{th}$  utterance, where  $c_n$  denotes the word numbers in the utterance  $u_n$  and  $w_i^{(n)}$  is the word embedding of the  $i_{th}$  word  $w_i^{(n)}$  in the utterance  $u_n$ .

We require utterance-level representations for probing where mixed synthesized information is present since the probing tasks rely on high-level reasoning. To get  $\mathbf{R}_{\text{history}}$  and  $\mathbf{R}_{\text{current}}$ , we independently averaged the representations corresponding to historical and contemporary utterances. Then

we concatenate the two averaged representations to get the final feature  $\mathbf{R}_{\text{probing}}$  for probing tasks, where the concatenation operation is denoted by  $\odot$ . The process is defined as follows:

$$\begin{aligned} \mathbf{R}_{\text{history}} &= \frac{1}{\sum_{t=1}^{n-1} c_t} \sum_{t=1}^{n-1} \mathbf{u}_t \\ &= \frac{1}{\sum_{t=1}^{n-1} c_t} \sum_{t=1}^{n-1} \sum_{i=1}^{c_t} \mathbf{w}_i^{(t)}, \end{aligned} \quad (1)$$

$$\mathbf{R}_{\text{current}} = \frac{1}{c_n} \sum_{i=1}^{c_n} \mathbf{w}_i^{(n)}, \quad (2)$$

$$\mathbf{R}_{\text{probing}} = \mathbf{R}_{\text{history}} \odot \mathbf{R}_{\text{current}}, \quad (3)$$

#### 3.2 Multi-Source Probing Approach

The prior MLP-based approach has been considered fairly intuitive to detect if the dialogue model captures pertinent information in the encoder states. However, it is difficult to effectively utilize the information encoded by the dialogue models using these approaches due to the divergent objectives of downstream probing tasks and dialogue generation during pre-training (Liu et al., 2021; Schick and Schütze, 2021).

To address this issue, we propose a **Multi-Source Probing (MSP)** method to probe the dialogue comprehension abilities of open-domain dialogue models. MSP conducts probing tasks in a generative manner, which is consistent with the pre-trained task to take full advantage of model capabilities. As various probing tasks require information from different aspects of the dialogue, which may differ greatly from the dialogue generation task, we propose a multi-source attention mechanism to aggregate features from multiple sources to accomplish diverse tasks.

Moreover, the application of potential understanding capabilities that might be encoded in the decoder parameters is also lacking in MLP-based probing methods, while MSP reapplies the discarded information and provides better modeling of contextual information. The overview of MSP is presented in Figure 1(c).

### 3.2.1 Continuous Prompt Learning

Continuous (or soft) prompt learning is one of the central parts in our MSP method, which is shown in Figure 1(b), as the generative approach of probing classification is considered to be more consistent with the goal in the pretraining phase, allowing the model to be more adaptable.

Specifically, we set the Prompt-Template as a sequence of different soft tokens for decoder input. Each soft token has a unique word embedding that is constantly adjusted and updated during the training stage. Consistent with the vanilla probing setting, the transformer decoder is finetuned in MSP, while the encoder parameters are fixed.

Since our probing task is based on classification, a verbalizer (Hu et al., 2022) class is constructed here to project the original probability distribution over the whole vocabulary to the set of label words given by the probing task.

### 3.2.2 Multi-Source Attention

Different downstream probing tasks have various focuses on the location of the required information (Sankar et al., 2019), while the general attention module of dialogue models tends to fail in locating and extracting information from multiple aspects and sources in a fine-grained way.

Therefore, to avoid overshadowing the key information contained in the dialogue context when the significance of information is unevenly distributed, we propose a multi-source attention mechanism, by using multiple cross-attention masks corresponding to different sources through the decoding process, to generate more reasonable attention that can extract relevant local features from different sources for probing classification.

As shown in Figure 2, the multi-source attention module takes turns allocating different cross-attention to different parts of encoded representations for soft prompt tokens. We adopt three types of attention masks to operate the multi-source cross attentions, which are separately called **history-source**, **current-source**, and **integrated-source**. Only the relevant portion of the dialogue is given

attention by the history-source and current-source attention functions. While the integrated-source cross-attention mask is built to gather information from the full context of the dialogue.

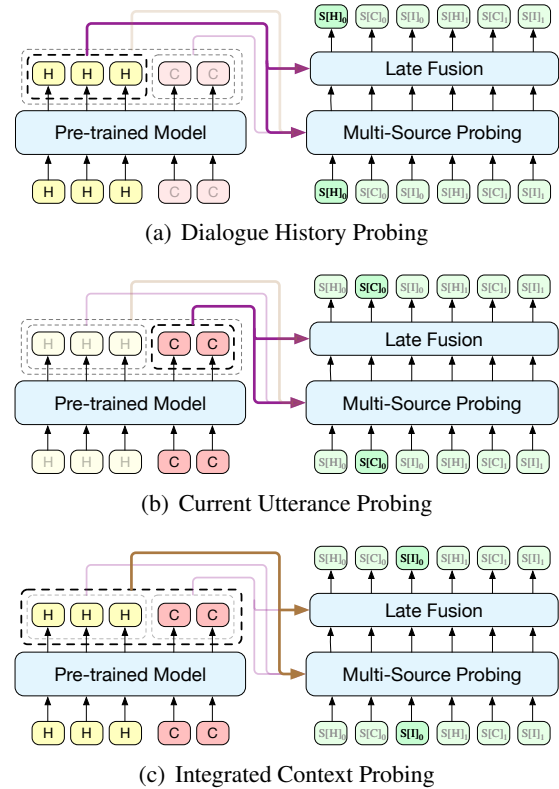


Figure 2: Overview of multi-source attention.

In Prompt-Template, the three consecutive tokens are grouped into one combined unit, which is designed to perform a round of coverage during the probing process. We assign the three soft tokens in the last combined unit as the prediction position where the averaged logit is passed to the subsequent verbalizer function, and then we can calculate the loss for the corresponding classes. This step is specially formulated to fuse the information captured by various soft-token representations that are focused independently on the history, current, and entire section of the input text.

The decoder  $f_\theta$  updates the hidden states  $\mathbf{H}_i$  conditioned on the past decoder states  $\mathbf{H}_{<i}$  with self-attention, the soft token embedding  $\mathbf{S}_i$ , as well as the encoded representations  $\mathbf{E}_k$  with cross-attention, where  $k$  could take values in  $\mathcal{K} = \{\text{history, current, integrated}\}$ , as follows:

$$\mathbf{H}_i = \begin{cases} f_\theta(\mathbf{H}_{<i}, \mathbf{S}_i, \mathbf{E}_{\text{his}}) & i = 0 \pmod{3} \\ f_\theta(\mathbf{H}_{<i}, \mathbf{S}_i, \mathbf{E}_{\text{cur}}) & i = 1 \pmod{3} \\ f_\theta(\mathbf{H}_{<i}, \mathbf{S}_i, \mathbf{E}_{\text{int}}) & i = 2 \pmod{3} \end{cases}. \quad (4)$$



### 3.2.3 Late Fusion

Previous studies (Vig, 2019; Vig and Belinkov, 2019) have shown that the attention mechanism is sensitive to local features, while global features of conversations should also be considered to produce a comprehensive representation.

Thus we introduce the late fusion module as a merging strategy for information integration to generate a comprehensive representation from encoder hidden states, which is combined with the probing decoder states after passing through an MLP layer with dropout. Late fusion is designed to capture the global information encoded by the language model which serves as the role of a complement to the probing decoder aimed at extracting precise local features, thus allowing a higher sensitivity to various features of the linguistic information.

Here we implement the late fusion by averaging the encoder hidden states  $\mathbf{E}_k$  that focus on the desired section of the text, where  $k$  could take  $\mathcal{K} = \{\text{history, current, integrated}\}$ . We would receive the representation  $\mathbf{L}_k$  after the late fusion, and  $\mathbf{A}_k$  is the final representation obtained by combining the decoder hidden states  $\mathbf{H}_k$  and the result from late fusion module.

$$\mathbf{L}_k = \text{MLP}(\text{avg}(\mathbf{E}_k)), \quad (5)$$

$$\mathbf{A}_k = \mathbf{H}_k + \mathbf{L}_k. \quad (6)$$

The probability of  $P_M(y|x)$  with probing label  $y$  is calculated as:

$$P_M(y|x) = \text{softmax}(\mathbf{V}(\text{avg}(\{\mathbf{A}_k^{(n)} | k \in \mathcal{K}\}))). \quad (7)$$

where  $\mathbf{V}$  is the language model head that projects the raw logits to the label space. Here  $n$  is the number of combined units in the soft token sequence.

## 4 Experiment

### 4.1 Probing Tasks

**TREC** (Li and Roth, 2002) is a question classification dataset consisting of questions labeled with relevant answer types. The task aims to determine the information category the question is requesting. **DialogueNLI** (Welleck et al., 2019) is a natural language inference task consisting of dialog turns with entailment, contradiction and neutral labels. **MultiWOZ** (Eric et al., 2020) is a multi-domain, goal-directed conversational dataset for exploring natural language comprehension.

**Schema-Guided Dialog dataset (SGD)** (Rastogi et al., 2020) is an intent-tracking task that requires reasoning over multiple turns of dialogue.

**SNIPS** (Coucke et al., 2018) is an intent classification task with crowd-sourced, single-turn queries labeled for intent.

**ScenarioSA** (Zhang et al., 2019) is a sentiment classification task with turn-level sentiment labels and inputs from multi-turn, open-ended dialogues.

**DailyDialog Topic** uses dialogues from the DailyDialog dataset (Li et al., 2017b) to create a probing task where the goal is to make inferences about the topic of conversations (Saleh et al., 2020).

### 4.2 Probing Methods

**Vanilla MLP-based Probing** adopts a two-layer Multi-Layer Perceptron classifier (Saleh et al., 2020; Parthasarathi et al., 2020), which takes the word embedding and encoder states corresponding to the conversation context of the pre-trained dialogue models as input.

**Prompt-based Probing** applies prompt learning as the principle for performing probing tasks. During training, only the embeddings of soft prompt tokens and the verbalizer parameters are fine-tuned while the pre-trained encoder and decoder parameters are fixed (Liu et al., 2021).

**Multi-Source Probing** is the proposed approach, which is characteristic of its multi-source attention and late fusion module. The parameters of the verbalizer, prompt token embeddings, and decoders are fine-tuned during training.

### 4.3 Open-domain Dialogue Models

We first train three widespread generative dialogue models for 20 epochs on DailyDialog dataset (Li et al., 2017b), using the Maximum-likelihood objective (Sutskever et al., 2014). Specifically, we train Transformer from scratch and fine-tune BlenderBot-small and DialoGPT-small with pre-trained parameters on DailyDialog for a fair comparison, as adopted in prior work (Saleh et al., 2020).

Besides, we adopt BlenderBot-medium [400M] and DialoGPT-medium [345M] without fine-tuning for the inspection of larger pre-trained models.

**Transformer** (Vaswani et al., 2017) is a typical language model with multiple attention layers. We implement it in the form of encoder-decoder structure, with a 2-layer encoder and a 2-layer decoder. **BlenderBot** (Roller et al., 2020) is a pre-trained dialogue model based on the encoder-decoder ar-

Method		TREC	DNLI	MWOZ	SGD	SNIPS	SSA	Topic
<b>Transformer</b>								
<b>MLP</b>	Emb.	83.1 <sup>*</sup> <sub>[1.07]</sub>	69.9 <sub>[0.79]</sub>	92.1 <sup>*</sup> <sub>[0.38]</sub>	69.9 <sup>*</sup> <sub>[0.55]</sub>	97.9 <sup>*</sup> <sub>[0.24]</sub>	77.1 <sub>[0.17]</sub>	53.3 <sub>[2.00]</sub>
	Enc.	79.7 <sub>[0.73]</sub>	68.7 <sub>[0.70]</sub>	91.3 <sub>[0.40]</sub>	67.0 <sub>[0.66]</sub>	96.8 <sub>[0.15]</sub>	77.8 <sup>*</sup> <sub>[0.36]</sub>	53.2 <sub>[1.35]</sub>
<b>PBP</b>	Emb.	68.2 <sub>[1.17]</sub>	52.0 <sub>[0.66]</sub>	57.0 <sup>*</sup> <sub>[0.30]</sub>	53.7 <sub>[0.56]</sub>	95.8 <sup>*</sup> <sub>[0.11]</sub>	64.6 <sup>*</sup> <sub>[0.25]</sub>	30.8 <sub>[1.15]</sub>
	Enc.	69.1 <sub>[1.14]</sub>	52.0 <sub>[0.64]</sub>	56.5 <sub>[0.26]</sub>	55.1 <sup>*</sup> <sub>[0.58]</sub>	94.9 <sub>[0.56]</sub>	63.5 <sub>[0.44]</sub>	35.1 <sup>*</sup> <sub>[1.31]</sub>
<b>MSP</b>	Emb.	90.5 <sub>[0.27]</sub>	75.0 <sub>[0.64]</sub>	95.2 <sub>[0.27]</sub>	78.9 <sub>[0.80]</sub>	97.9 <sub>[0.11]</sub>	79.4 <sub>[0.28]</sub>	60.7 <sub>[1.19]</sub>
	Enc.	<b>91.2</b> <sub>[0.50]</sub>	<b>76.2</b> <sub>[0.23]</sub>	<b>95.4</b> <sub>[0.21]</sub>	<b>80.2</b> <sub>[0.48]</sub>	<b>98.3</b> <sub>[0.21]</sub>	<b>80.7</b> <sub>[0.29]</sub>	<b>63.1</b> <sub>[1.61]</sub>
<b>BlenderBot<sub>SMALL</sub></b>								
<b>MLP</b>	Emb.	85.7 <sub>[0.69]</sub>	72.4 <sub>[0.23]</sub>	92.5 <sub>[0.18]</sub>	72.9 <sub>[0.26]</sub>	97.5 <sub>[0.27]</sub>	78.7 <sub>[0.37]</sub>	52.2 <sub>[2.64]</sub>
	Enc.	92.1 <sup>*</sup> <sub>[0.56]</sub>	84.3 <sup>*</sup> <sub>[0.29]</sub>	93.6 <sup>*</sup> <sub>[0.32]</sub>	76.0 <sup>*</sup> <sub>[0.48]</sub>	97.7 <sub>[0.18]</sub>	83.2 <sup>*</sup> <sub>[0.16]</sub>	65.2 <sup>*</sup> <sub>[0.78]</sub>
<b>PBP</b>	Emb.	75.5 <sub>[2.18]</sub>	54.5 <sub>[1.01]</sub>	58.9 <sub>[0.67]</sub>	56.4 <sub>[1.28]</sub>	95.6 <sub>[0.64]</sub>	62.9 <sub>[0.85]</sub>	50.8 <sub>[2.12]</sub>
	Enc.	92.1 <sup>*</sup> <sub>[0.97]</sub>	61.0 <sup>*</sup> <sub>[3.30]</sub>	81.9 <sup>*</sup> <sub>[6.16]</sub>	80.6 <sup>*</sup> <sub>[2.74]</sub>	97.3 <sup>*</sup> <sub>[0.65]</sub>	76.3 <sup>*</sup> <sub>[0.76]</sub>	66.7 <sup>*</sup> <sub>[0.94]</sub>
<b>MSP</b>	Emb.	91.0 <sub>[0.64]</sub>	77.9 <sub>[1.00]</sub>	95.7 <sub>[0.15]</sub>	82.0 <sub>[1.34]</sub>	98.1 <sub>[0.19]</sub>	80.3 <sub>[0.26]</sub>	60.5 <sub>[1.54]</sub>
	Enc.	<b>93.8</b> <sub>[0.70]</sub>	<b>87.4</b> <sub>[0.25]</sub>	<b>96.5</b> <sub>[0.30]</sub>	<b>88.8</b> <sub>[0.55]</sub>	<b>98.6</b> <sub>[0.16]</sub>	<b>84.4</b> <sub>[0.33]</sub>	<b>71.3</b> <sub>[1.71]</sub>
<b>DialoGPT<sub>SMALL</sub></b>								
<b>MLP</b>	Emb.	88.4 <sub>[0.63]</sub>	73.8 <sub>[0.69]</sub>	92.9 <sup>*</sup> <sub>[0.29]</sub>	73.4 <sup>*</sup> <sub>[0.42]</sub>	98.8 <sub>[0.14]</sub>	80.0 <sub>[0.32]</sub>	52.6 <sub>[1.22]</sub>
	Enc.	92.1 <sup>*</sup> <sub>[0.28]</sub>	80.9 <sup>*</sup> <sub>[0.30]</sub>	90.8 <sub>[0.23]</sub>	72.8 <sub>[0.24]</sub>	98.7 <sub>[0.09]</sub>	81.1 <sup>*</sup> <sub>[0.23]</sub>	65.2 <sup>*</sup> <sub>[1.31]</sub>
<b>PBP</b>	Emb.	37.5 <sub>[3.23]</sub>	42.3 <sub>[1.27]</sub>	28.2 <sub>[1.99]</sub>	10.6 <sub>[0.52]</sub>	71.1 <sub>[2.92]</sub>	57.1 <sub>[0.01]</sub>	26.0 <sub>[0.00]</sub>
	Enc.	52.0 <sup>*</sup> <sub>[1.17]</sub>	50.0 <sup>*</sup> <sub>[0.61]</sub>	44.9 <sup>*</sup> <sub>[0.60]</sub>	34.3 <sup>*</sup> <sub>[1.02]</sub>	86.2 <sup>*</sup> <sub>[0.62]</sub>	57.9 <sup>*</sup> <sub>[0.32]</sub>	36.7 <sup>*</sup> <sub>[1.12]</sub>
<b>MSP</b>	Emb.	94.2 <sub>[0.38]</sub>	<b>83.7</b> <sub>[0.39]</sub>	96.1 <sub>[0.27]</sub>	85.4 <sub>[0.03]</sub>	98.9 <sub>[0.15]</sub>	82.3 <sub>[0.46]</sub>	58.7 <sub>[0.64]</sub>
	Enc.	<b>96.8</b> <sub>[0.63]</sub>	83.4 <sub>[0.21]</sub>	<b>96.2</b> <sub>[0.62]</sub>	<b>85.7</b> <sub>[0.43]</sub>	<b>99.0</b> <sub>[0.64]</sub>	<b>82.4</b> <sub>[0.37]</sub>	<b>66.7</b> <sub>[0.66]</sub>

Table 1: Accuracy on probing tasks for **Transformer**, **BlenderBot<sub>SMALL</sub>**, **DialoGPT<sub>SMALL</sub>**. Experiments were conducted on three probing methods: 1) **MLP**: Vanilla MLP-based Probing 2) **PBP**: Prompt-based Probing. 3) **MSP**: Multi-Source Probing. Best results are marked in **bold**, and data that passed the significance test (t-test, p-value < 0.05) between word embeddings and encoder states under one specific probing method are super-scripted with an asterisk \*. The numbers in square brackets represent the standard deviation.

chitecture which is first pre-trained on 1.5B Reddit comment threads (Baumgartner et al., 2020) and later fine-tuned on Blended SkillTalk (BST) dataset (Smith et al., 2020).

**DialoGPT** (Zhang et al., 2020) is a dialogue response generation model for multi-turn conversations with a single decoder, which is pre-trained on large-scale Reddit data (Baumgartner et al., 2020).

## 5 Analysis

We will detail experimental results in this section, including the analysis of our main experiments on the performance of different probing methods, the ablation study of Multi-Source Probing architectures, and evaluations of different dialogue models and experimental settings.

### 5.1 Main Results

The main results are presented in Table 1, where each probing task is evaluated by calculating an average score of accuracy. We analyze the results from the following perspectives:

**Comparison between methods:** For all probing tasks and dialogue models, MSP achieves the best performances, indicating that our method more effectively leverages the relevant information encoded in the intermediate representations to conduct probing tasks. Besides, the majority of encoder state results outperform word embedding results, indicating that encoder states contain more semantic features than word embeddings. Thus, to some extent, dialogue models learn semantic information from conversations, which is required in conversational understanding tasks.

For the MLP-based probing method, we observe a similar phenomenon of the Transformer model that the performances of encoder states are not superior in many tasks as reported in prior work (Saleh et al., 2020). This observation demonstrates that the vanilla probing method has limitations in utilizing encoder states to conduct conversational understanding tasks, due to the gap between downstream classification tasks and the pre-trained dialogue generation task (Liu et al., 2021; Schick

Method		TREC	DNLI	MWOZ	SGD	SNIPS	SSA	Topic
<b>BlenderBot<sub>MEDIUM</sub></b>								
<b>MLP</b>	Emb.	83.9 <sub>[1.10]</sub>	73.3 <sub>[0.77]</sub>	90.2 <sub>[0.51]</sub>	65.3 <sub>[0.80]</sub>	98.3 <sub>[0.35]</sub>	79.3 <sub>[0.28]</sub>	45.3 <sub>[3.59]</sub>
	Enc.	91.1 <sub>[0.20]</sub>	85.1 <sub>[0.35]</sub>	90.6 <sub>[0.36]</sub>	66.3 <sub>[0.46]</sub>	98.1 <sub>[0.12]</sub>	83.6 <sub>[0.45]</sub>	56.2 <sub>[1.20]</sub>
<b>MSP</b>	Emb.	92.6 <sub>[0.27]</sub>	79.1 <sub>[0.52]</sub>	96.3 <sub>[0.31]</sub>	82.8 <sub>[0.46]</sub>	98.1 <sub>[0.66]</sub>	82.1 <sub>[0.53]</sub>	61.5 <sub>[0.51]</sub>
	Enc.	<b>94.1</b> <sub>[0.44]</sub>	<b>90.3</b> <sub>[0.23]</sub>	<b>97.4</b> <sub>[0.65]</sub>	<b>91.3</b> <sub>[0.48]</sub>	<b>98.9</b> <sub>[0.61]</sub>	<b>87.3</b> <sub>[0.50]</sub>	<b>72.3</b> <sub>[1.09]</sub>
<b>DialoGPT<sub>MEDIUM</sub></b>								
<b>MLP</b>	Emb.	89.2 <sub>[0.87]</sub>	74.5 <sub>[0.55]</sub>	92.9 <sub>[0.46]</sub>	74.6 <sub>[0.58]</sub>	98.4 <sub>[0.11]</sub>	80.4 <sub>[0.35]</sub>	49.6 <sub>[2.47]</sub>
	Enc.	94.8 <sub>[0.13]</sub>	84.7 <sub>[0.28]</sub>	91.5 <sub>[0.20]</sub>	76.0 <sub>[0.53]</sub>	98.6 <sub>[0.12]</sub>	80.9 <sub>[0.30]</sub>	67.0 <sub>[0.94]</sub>
<b>MSP</b>	Emb.	95.2 <sub>[0.53]</sub>	88.2 <sub>[0.57]</sub>	<b>96.5</b> <sub>[0.44]</sub>	86.1 <sub>[0.49]</sub>	99.0 <sub>[0.63]</sub>	82.8 <sub>[0.53]</sub>	59.7 <sub>[0.32]</sub>
	Enc.	<b>97.4</b> <sub>[0.34]</sub>	<b>89.5</b> <sub>[0.52]</sub>	96.4 <sub>[0.63]</sub>	<b>89.6</b> <sub>[0.38]</sub>	<b>99.1</b> <sub>[0.42]</sub>	<b>85.7</b> <sub>[0.80]</sub>	<b>70.2</b> <sub>[0.92]</sub>

Table 2: The performance of MLP and MSP on two large-scale pre-trained dialogue models **BlenderBot<sub>MEDIUM</sub>** and **DialoGPT<sub>MEDIUM</sub>**. Best results are marked in **bold**, and data that passed the significance test (t-test, p-value < 0.05) are super-scripted with an asterisk \*. The numbers in square brackets represent the standard deviation.

and Schütze, 2021). The prompt-based probing approach obtains the worst performance among the three probing methods. Although it performs the probing tasks in a generative manner, this approach cannot effectively extract relevant features required in conversational understanding tasks, leading to undesirable results.

**Comparison between models:** As the parameters of dialogue models increase, the performance on dialogue understanding tasks also improves based on the MSP method. However, DialoGPT doesn’t outperform BlenderBot on some tasks and the performances of encoder states are not always significantly better than those of word embeddings. This is probably because DialoGPT adopts a single decoder structure with the uni-directional attention mechanism, which encodes partial context (content before each token) information for tokens, leading to asymmetric representations of dialogue history and current utterances. By contrast, the encoder-decoder-based BlenderBot applies the bi-directional attention mechanism to encode bi-directional information (content before and after each token) for tokens, achieving more consistent and superior performances on conversational understanding tasks.

We also evaluated the comprehension ability of pre-trained language models such as BERT, BART, and T5 through MSP, and the details are attached in Appendix 6.

## 5.2 Ablation Study

Here we set up a series of ablation experiments to investigate the validity and necessity of the components of our Multi-Source Probing approach, which is composed of two main parts: the multi-source

attention mechanism and the late fusion module. The multi-source attention mechanism is designed to pay fair attention to both the historical and current turns of the conversational context, while the design of the late fusion module is motivated by the fact that many probing tasks need to capture global features while maintaining a high sensitivity to local features, so we add a layer of late fusion after the decoder for fusing global and local features in the final representation. Due to the length limit, the results are presented in Table 4 in the Appendix.

### 5.2.1 Ablation Setting

Several sets of ablation experiments are designed to verify the necessity and effectiveness of different modules:

**MSP w/o LF** is an ablation setting where the late fusion module is removed compared to the complete MSP method.

**MSP w/o MS** is an ablation setting where the multi-source attention and late fusion module are removed. This approach has the same architecture as the prompt-based probing method except that the decoder is fine-tuned during training.

### 5.2.2 Effect of Late Fusion Module

In our experiment settings, we set ablation experiments on the effectiveness of late-fusion. Through the analysis, we discover that the late-fusion architecture enables a significant improvement in the ability of the probing model to extract the representational information encoded by the pre-trained language model. Furthermore, it not only incorporates the information provided by the encoder very well but also provides a nuanced insight for our probing architecture of representational informa-

Method		TREC	DNLI	MWOZ	SGD	SNIPS	SSA	Topic
<b>Transformer</b>								
<b>MLP</b>	Rand.Emb.	71.8 <sub>[1.73]</sub>	60.4 <sub>[0.42]</sub>	87.2 <sub>[0.50]</sub>	60.5 <sub>[0.60]</sub>	96.6 <sub>[0.36]</sub>	70.6 <sub>[0.29]</sub>	39.8 <sub>[2.49]</sub>
	Orig.Emb.	83.1 <sub>[1.07]</sub>	69.9 <sub>[0.79]</sub> *	92.1 <sub>[0.38]</sub>	69.9 <sub>[0.55]</sub> *	97.9 <sub>[0.24]</sub> *	77.1 <sub>[0.17]</sub> *	53.3 <sub>[2.00]</sub>
	Shuf.Emb.	83.0 <sub>[1.08]</sub> *	69.8 <sub>[0.39]</sub> *	92.0 <sub>[0.46]</sub> *	69.8 <sub>[0.38]</sub> *	97.9 <sub>[0.35]</sub> *	77.3 <sub>[0.43]</sub>	53.5 <sub>[1.44]</sub>
	Shuf.Enc.	77.2 <sub>[0.66]</sub>	68.8 <sub>[0.28]</sub>	90.1 <sub>[0.27]</sub>	67.6 <sub>[0.35]</sub>	96.6 <sub>[0.36]</sub>	77.1 <sub>[0.32]</sub>	55.9 <sub>[2.17]</sub> *
<b>MSP</b>	Rand.Emb.	80.4 <sub>[0.72]</sub>	65.5 <sub>[0.34]</sub>	94.2 <sub>[0.41]</sub>	75.9 <sub>[1.12]</sub>	97.5 <sub>[0.21]</sub>	75.6 <sub>[1.02]</sub>	37.2 <sub>[1.76]</sub>
	Orig.Emb.	90.5 <sub>[0.27]</sub> *	75.0 <sub>[0.64]</sub> *	95.2 <sub>[0.27]</sub> *	78.9 <sub>[0.80]</sub> *	97.9 <sub>[0.11]</sub> *	79.4 <sub>[0.28]</sub> *	60.7 <sub>[1.19]</sub> *
	Shuf.Emb.	84.2 <sub>[0.67]</sub>	68.9 <sub>[0.41]</sub>	94.9 <sub>[0.79]</sub>	76.9 <sub>[0.67]</sub>	98.1 <sub>[0.25]</sub>	77.7 <sub>[1.43]</sub>	52.3 <sub>[1.44]</sub>
	Shuf.Enc.	84.9 <sub>[0.59]</sub>	70.0 <sub>[0.24]</sub> *	94.7 <sub>[0.53]</sub>	78.1 <sub>[0.23]</sub> *	98.3 <sub>[0.32]</sub> *	78.8 <sub>[0.95]</sub> *	57.5 <sub>[1.76]</sub> *
<b>BlenderBot<sub>SMALL</sub></b>								
<b>MLP</b>	Rand.Emb.	76.2 <sub>[1.10]</sub>	63.3 <sub>[0.79]</sub>	90.6 <sub>[0.48]</sub>	65.8 <sub>[0.54]</sub>	97.2 <sub>[0.29]</sub>	73.1 <sub>[0.44]</sub>	40.1 <sub>[2.12]</sub>
	Orig.Emb.	85.7 <sub>[0.69]</sub> *	72.4 <sub>[0.23]</sub> *	92.5 <sub>[0.18]</sub> *	72.9 <sub>[0.26]</sub> *	97.5 <sub>[0.27]</sub> *	78.7 <sub>[0.37]</sub> *	52.2 <sub>[2.64]</sub> *
	Shuf.Emb.	85.3 <sub>[0.60]</sub> *	72.7 <sub>[0.51]</sub>	92.5 <sub>[0.25]</sub> *	72.9 <sub>[0.28]</sub> *	97.4 <sub>[0.20]</sub>	78.8 <sub>[0.43]</sub>	51.7 <sub>[2.60]</sub>
	Shuf.Enc.	83.1 <sub>[1.34]</sub>	79.8 <sub>[0.22]</sub> *	89.7 <sub>[0.23]</sub>	68.0 <sub>[0.44]</sub>	97.5 <sub>[0.22]</sub>	78.9 <sub>[0.20]</sub>	59.9 <sub>[0.78]</sub> *
<b>MSP</b>	Rand.Emb.	84.0 <sub>[0.82]</sub>	74.3 <sub>[0.25]</sub>	94.6 <sub>[0.50]</sub>	82.2 <sub>[0.53]</sub>	97.6 <sub>[0.16]</sub>	76.4 <sub>[0.75]</sub>	47.7 <sub>[1.77]</sub>
	Orig.Emb.	91.0 <sub>[0.64]</sub> *	77.9 <sub>[1.00]</sub> *	95.7 <sub>[0.15]</sub> *	82.0 <sub>[1.34]</sub>	98.1 <sub>[0.19]</sub> *	80.3 <sub>[0.26]</sub> *	60.5 <sub>[1.54]</sub> *
	Shuf.Emb.	88.7 <sub>[0.66]</sub> *	75.8 <sub>[0.31]</sub>	95.8 <sub>[0.29]</sub>	83.9 <sub>[0.26]</sub>	98.1 <sub>[0.09]</sub>	79.8 <sub>[0.43]</sub>	56.3 <sub>[1.50]</sub>
	Shuf.Enc.	85.2 <sub>[0.45]</sub>	84.6 <sub>[0.28]</sub> *	95.4 <sub>[0.46]</sub>	84.7 <sub>[0.12]</sub> *	98.3 <sub>[0.15]</sub> *	81.9 <sub>[0.24]</sub> *	65.9 <sub>[4.16]</sub> *
<b>DialoGPT<sub>SMALL</sub></b>								
<b>MLP</b>	Rand.Emb.	75.3 <sub>[0.55]</sub>	74.1 <sub>[0.61]</sub>	91.6 <sub>[0.27]</sub>	68.2 <sub>[0.48]</sub>	96.6 <sub>[0.19]</sub>	75.3 <sub>[0.44]</sub>	42.2 <sub>[2.18]</sub>
	Orig.Emb.	88.4 <sub>[0.63]</sub> *	73.8 <sub>[0.69]</sub>	92.9 <sub>[0.29]</sub>	73.4 <sub>[0.42]</sub>	98.8 <sub>[0.14]</sub> *	80.0 <sub>[0.32]</sub> *	52.6 <sub>[1.22]</sub> *
	Shuf.Emb.	89.5 <sub>[0.84]</sub> *	73.9 <sub>[0.66]</sub>	93.0 <sub>[0.31]</sub> *	73.3 <sub>[0.20]</sub> *	98.9 <sub>[0.17]</sub> *	80.2 <sub>[0.27]</sub> *	50.8 <sub>[1.62]</sub>
	Shuf.Enc.	80.1 <sub>[0.59]</sub>	76.7 <sub>[0.20]</sub> *	86.9 <sub>[0.58]</sub>	66.8 <sub>[0.16]</sub>	96.7 <sub>[0.24]</sub>	76.0 <sub>[0.23]</sub>	65.4 <sub>[0.32]</sub> *
<b>MSP</b>	Rand.Emb.	87.8 <sub>[0.32]</sub>	78.6 <sub>[0.76]</sub>	92.3 <sub>[0.26]</sub>	70.4 <sub>[1.45]</sub>	97.1 <sub>[0.12]</sub>	77.4 <sub>[0.56]</sub>	52.2 <sub>[0.87]</sub>
	Orig.Emb.	94.2 <sub>[0.38]</sub> *	83.7 <sub>[0.39]</sub> *	96.1 <sub>[0.27]</sub> *	85.4 <sub>[0.03]</sub> *	98.9 <sub>[0.15]</sub> *	82.3 <sub>[0.46]</sub> *	58.7 <sub>[0.64]</sub> *
	Shuf.Emb.	85.4 <sub>[0.28]</sub>	81.2 <sub>[0.24]</sub>	94.5 <sub>[0.19]</sub>	81.5 <sub>[1.02]</sub>	98.5 <sub>[0.12]</sub>	80.3 <sub>[0.39]</sub>	53.6 <sub>[4.21]</sub>
	Shuf.Enc.	91.5 <sub>[0.63]</sub> *	82.7 <sub>[0.26]</sub> *	94.6 <sub>[0.21]</sub>	82.3 <sub>[0.34]</sub> *	98.8 <sub>[0.05]</sub> *	81.5 <sub>[0.21]</sub> *	66.0 <sub>[2.18]</sub> *

Table 3: The performance with random embeddings and shuffled order of words in dialogue context. Data that passed the significance test (t-test, p-value < 0.05) are super-scripted with an asterisk \*.

tion at the shallow level. Consequently, this makes the structure more hierarchical and efficient in integrating different depths of encoded information.

We found from the results of our experiments that when the Multi-Source Probing Method does not have a late-fusion layer, the probing results of encoder states in several tasks are not better than those of word embeddings, which indicates that when we discard the late fusion, we also discard some classification local features needed for the task. We found that the ground-truth labels for these tasks are often determined by a combination of certain keywords in the historical utterance and some obvious prompt words in the current utterance of the dialog, so if we simply use the representations generated by the multi-layer transformer in the decoder, this tends to draw out only the global features and ignore the involvement of local features. So in this step of the experiment, we verified the significant role of the late fusion module for

synthesizing local features.

### 5.2.3 Effect of Multi-Source Attention

We further developed ablation experiments to explore the impact of discarding the multi-source attention module of our MSP method, where an exciting discovery is found that the accuracy of the probing task was reduced by more than 10% when the MSP method did not efficiently utilize the multi-source attention module, suggesting that we need to introduce specialized designs to focus on different parts of the input text when evaluating the ability of a language model to understand a conversation. This demonstrates the need to introduce specialized designs to focus on different parts of the input text, which is a concern that generative dialogue models are constantly focusing on during the pre-training process. We experimentally found that a single cross-attention module is not effective in accomplishing our probing purpose, and thus



a multiple-source attention mechanism is a very central component of our probing approach.

### 5.3 Extension Experiments

Considering the undesirable performance of the prompt-based probing approach as shown in Table 1, we adopt MSP and MLP methods in the extension experiments.

#### 5.3.1 Impact of Model Scale

Table 2 shows the performance of large-scale pre-trained conversational models on probing tasks. As can be seen, with the increase of model parameters, the performance on dialogue comprehension tasks also increases. The pre-trained dialogue models demonstrate a strong capability of conversational understanding even without fine-tuning on downstream dialogue corpus. For the MLP-based probing method, there is no obvious difference between the performance of encoder states and word embeddings in many tasks. By contrast, our approach is applicable to models of different scales from the 2-layer Transformer trained from scratch to the large-scale pre-trained BlenderBot and DialoGPT.

#### 5.3.2 Impact of Word Embedding

During the training process of dialogue generation, word embeddings can learn and encode linguistic knowledge of conversations (Ravichander et al., 2020) as Word2Vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014). Thus, we substitute the trained word embeddings with randomly initialized ones and conduct probing experiments to investigate the impact of word embeddings.

The performance with random embeddings is shown in Table 3. As we can see, there is a significant gap between the performance of random word embeddings and original ones, indicating that the trained word embeddings encode semantic information required by conversational understanding. In addition, the performance with random embeddings achieves above 90% on some tasks, such as SNIPS and MWOZ, while only obtaining less than 70% on others. It shows that different tasks require different degrees of conversational semantics.

#### 5.3.3 Impact of Dialogue Structure

To examine whether dialogue models leverage dialogue structure in conversational understanding, we shuffle the order of the input tokens within dialogue history and the current utterance respectively. The results are shown in Table 3. We note that

the MLP method weakens the feature of the word order by the average pooling operation, while MSP offers superior modeling of the word order.

We observed that the performances of both word embeddings and encoder states decrease substantially with shuffled input. Furthermore, the performance gap is even larger for encoder states, indicating that dialogue models can capture the context and flow of the dialogue for conversational understanding, rather than just processing individual words or utterances in isolation.

## 6 Conclusion

In this paper, we propose a Multi-Source Probing (MSP) method to probe the dialogue comprehension abilities of open-domain dialogue models. It conducts probing tasks in a generative manner that is consistent with the pre-training task of dialogue models. Besides, we propose the multi-source attention mechanism to aggregate features from multiple sources and the late fusion module to capture global features for downstream tasks. Our experimental results indicate the validity and reliability of the MSP method, which could also offer insight into the impact of the model scale, embedding quality, and dialogue structure on the conversational understanding capability of dialogue models when particular experimental settings are presented. This research underscores the importance of a comprehensive probing framework for dialogue models and paves the way for future studies aimed at enhancing their understanding capabilities.

### Limitations

Although the Multi-Source Probing (MSP) method can precisely detect the conversational understanding of open-domain dialogue models of different scales, it still faces two limitations. First, we focus on evaluating three widespread dialogue models in our experiments due to the limitation of computational resources. Dialogue models of different structures and scales could be probed with MSP in future work. Second, we adopt several representative classification tasks as our probing tasks, following previous work (Saleh et al., 2020). These tasks require different dialogue comprehension skills and have different degrees of difficulty, as analyzed in Section 5.3.2. In future work, a wide range of tasks of different complexity in different domains could be conducted based on MSP to construct a benchmark of conversational understanding.

## References

- Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.
- Afra Alishahi, Marie Barking, and Grzegorz Chrupała. 2017. Encoding of phonology in a recurrent neural model of grounded speech. *arXiv preprint arXiv:1706.03815*.
- Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. Probing linguistic features of sentence-level representations in neural relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1534–1545.
- Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. Plato: Pre-trained dialogue generation model with discrete latent variable. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 85–96.
- Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 830–839.
- Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? *arXiv preprint arXiv:1704.03471*.
- Yonatan Belinkov and James Glass. 2017. [Analyzing hidden representations in end-to-end automatic speech recognition systems](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2018. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. *arXiv preprint arXiv:1801.07772*.
- Meriem Beloucif and Chris Biemann. 2021. Probing pre-trained language models for semantic attributes and their values. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2554–2559.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Itsugun Cho and Hiroaki Saito. 2021. Anna: A dapper open-domain dialogue agent based on a joint attention network. *Journal of Natural Language Processing*, 28(4):1184–1209.
- Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. [What you can cram into a single  \$\&\&!##^\*\$  vector: Probing sentence embeddings for linguistic properties](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.
- Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.
- Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, and Stephan Vogel. 2017. Understanding and improving morphological learning in the neural machine translation decoder. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 142–151.
- Abhishek Das, Federico Carnevale, Hamza Merzic, Laura Rimell, Rosalia Schneider, Josh Abramson, Alden Hung, Arun Ahuja, Stephen Clark, Gregory Wayne, et al. 2020. Probing emergent semantics in predictive agents via question answering. *Proceedings of the 37th International Conference on Machine Learning*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2020. The second conversational intelligence challenge (convai2). In *The NeurIPS’18 Competition*, pages 187–208. Springer.
- Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. [MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.
- Karen Hambardzumyan, Hrant Khachatryan, and Jonathan May. 2021. [Warp: Word-level adversarial reprogramming](#).

- Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. [Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification](#).
- Tianbo Ji, Yvette Graham, Gareth JF Jones, Chenyang Lyu, and Qun Liu. 2022. Achieving reliable human assessment of open-domain dialogue systems. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6416–6437.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8460–8478.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.
- Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017a. Adversarial learning for neural dialogue generation. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169.
- Xin Li and Dan Roth. 2002. [Learning question classifiers](#). In *COLING 2002: The 19th International Conference on Computational Linguistics*.
- Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017b. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Prasanna Parthasarathi, Joelle Pineau, and Sarath Chandar. 2020. How to evaluate your dialogue system: Probe tasks as an alternative for token-level evaluation metrics. *arXiv preprint arXiv:2008.10427*.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.
- Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayan-deh, Lars Liden, and Jianfeng Gao. 2020. Soloist: Few-shot task-oriented dialog with a single pre-trained auto-regressive model. *arXiv preprint arXiv:2005.05298*.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.
- Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2020. Probing the probing paradigm: Does probing accuracy entail task relevance? *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*.
- Kyle Richardson, Hai Hu, Lawrence Moss, and Ashish Sabharwal. 2020. [Probing natural language inference models through semantic fragments](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8713–8721.
- Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. 2020. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.
- Abdelrhman Saleh, Tovly Deutsch, Stephen Casper, Yonatan Belinkov, and Stuart Shieber. 2020. [Probing neural dialog models for conversational understanding](#). *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 132–143.
- Chinnadhurai Sankar, Sandeep Subramanian, Christopher Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do neural dialog systems use the conversation history effectively? an empirical study. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 32–37.
- Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269.

- Iulian Serban, Alessandro Sordani, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.
- Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. [Can you put it all together: Evaluating conversational agents' ability to blend skills](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2021–2030, Online. Association for Computational Linguistics.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.
- Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2018. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics System Demonstrations*, page 37.
- Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76.
- Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019a. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019b. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*.
- Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. 2020. Tod-bert: Pre-trained natural language understanding for task-oriented dialogue. *arXiv preprint arXiv:2004.06871*.
- Zequ Wu, Michel Galley, Chris Brockett, Yizhe Zhang, Xiang Gao, Chris Quirk, Rik Koncel-Kedziorski, Jianfeng Gao, Hannaneh Hajishirzi, Mari Ostendorf, et al. 2021. A controllable model of grounded response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14085–14093.
- Kong Yawei and Kai Fan. 2021. Probing multi-modal machine translation with pre-trained language model. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3689–3699.
- Yazhou Zhang, Lingling Song, Dawei Song, Peng Guo, Junwei Zhang, and Peng Zhang. 2019. Scenarios: A large scale conversational database for interactive sentiment analysis. *arXiv preprint arXiv:1907.05562*.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278.
- Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3377–3390.
- Peixiang Zhong, Chen Zhang, Hao Wang, Yong Liu, and Chunyan Miao. 2020. Towards persona-based empathetic conversational models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6556–6566.
- Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629.



Method		TREC	DNLI	MWOZ	SGD	SNIPS	SSA	Topic
<b>Transformer</b>								
<b>MSP</b>	Emb.	90.5 <sub>[0.27]</sub>	75.0 <sub>[0.64]</sub>	95.2 <sub>[0.27]</sub>	78.9 <sub>[0.80]</sub>	97.9 <sub>[0.11]</sub>	79.4 <sub>[0.28]</sub>	60.7 <sub>[1.19]</sub>
	Enc.	<b>91.2</b> <sub>[0.50]</sub> *	<b>76.2</b> <sub>[0.23]</sub> *	<b>95.4</b> <sub>[0.21]</sub> *	<b>80.2</b> <sub>[0.48]</sub> *	<b>98.3</b> <sub>[0.21]</sub> *	<b>80.7</b> <sub>[0.29]</sub> *	<b>63.1</b> <sub>[1.61]</sub> *
<b>MSP w/o LF</b>	Emb.	90.0 <sub>[0.54]</sub>	75.1 <sub>[0.40]</sub>	95.2 <sub>[0.19]</sub>	78.7 <sub>[0.58]</sub>	97.9 <sub>[0.42]</sub>	80.1 <sub>[0.22]</sub>	61.6 <sub>[2.17]</sub>
	Enc.	90.6 <sub>[0.38]</sub>	75.4 <sub>[1.05]</sub>	95.2 <sub>[0.21]</sub>	79.7 <sub>[0.54]</sub>	98.2 <sub>[0.23]</sub>	79.4 <sub>[0.42]</sub>	62.1 <sub>[1.65]</sub>
<b>MSP w/o MS</b>	Emb.	87.7 <sub>[0.30]</sub>	68.3 <sub>[0.67]</sub>	84.4 <sub>[0.35]</sub>	71.6 <sub>[0.73]</sub>	97.7 <sub>[0.41]</sub>	73.2 <sub>[0.50]</sub>	60.5 <sub>[1.65]</sub>
	Enc.	89.1 <sub>[0.30]</sub> *	69.2 <sub>[0.93]</sub>	83.9 <sub>[0.59]</sub>	70.2 <sub>[0.75]</sub>	97.7 <sub>[0.11]</sub>	72.8 <sub>[0.30]</sub>	61.9 <sub>[1.37]</sub>
<b>BlenderBot<sub>SMALL</sub></b>								
<b>MSP</b>	Emb.	91.0 <sub>[0.64]</sub>	77.9 <sub>[1.00]</sub>	95.7 <sub>[0.15]</sub>	82.0 <sub>[1.34]</sub>	98.1 <sub>[0.19]</sub>	80.3 <sub>[0.26]</sub>	60.5 <sub>[1.54]</sub>
	Enc.	<b>93.8</b> <sub>[0.70]</sub> *	<b>87.4</b> <sub>[0.25]</sub> *	<b>96.5</b> <sub>[0.30]</sub> *	<b>88.8</b> <sub>[0.55]</sub> *	<b>98.6</b> <sub>[0.16]</sub> *	<b>84.4</b> <sub>[0.33]</sub> *	<b>71.3</b> <sub>[1.71]</sub> *
<b>MSP w/o LF</b>	Emb.	90.4 <sub>[1.07]</sub>	74.7 <sub>[0.30]</sub>	95.5 <sub>[0.19]</sub>	81.9 <sub>[0.67]</sub>	98.2 <sub>[0.23]</sub>	79.8 <sub>[0.47]</sub>	60.4 <sub>[1.44]</sub>
	Enc.	93.0 <sub>[0.62]</sub>	85.5 <sub>[0.39]</sub>	96.4 <sub>[0.32]</sub>	88.8 <sub>[0.86]</sub>	98.0 <sub>[0.31]</sub>	84.2 <sub>[0.41]</sub>	71.2 <sub>[0.27]</sub>
<b>MSP w/o MS</b>	Emb.	90.5 <sub>[0.79]</sub>	71.7 <sub>[0.59]</sub>	90.2 <sub>[0.94]</sub>	81.2 <sub>[0.17]</sub>	97.9 <sub>[0.29]</sub>	74.0 <sub>[0.56]</sub>	58.4 <sub>[2.62]</sub>
	Enc.	93.4 <sub>[0.25]</sub> *	84.7 <sub>[0.59]</sub> *	95.8 <sub>[0.21]</sub> *	87.9 <sub>[0.30]</sub> *	98.5 <sub>[0.06]</sub> *	83.1 <sub>[0.17]</sub> *	70.7 <sub>[2.95]</sub> *
<b>DialogPT<sub>SMALL</sub></b>								
<b>MSP</b>	Emb.	94.2 <sub>[0.38]</sub>	<b>83.7</b> <sub>[0.39]</sub> *	96.1 <sub>[0.27]</sub>	85.4 <sub>[0.03]</sub>	98.9 <sub>[0.15]</sub>	82.3 <sub>[0.46]</sub>	58.7 <sub>[0.64]</sub>
	Enc.	<b>96.8</b> <sub>[0.63]</sub> *	83.4 <sub>[0.21]</sub>	<b>96.2</b> <sub>[0.62]</sub> *	<b>85.7</b> <sub>[0.43]</sub> *	<b>99.0</b> <sub>[0.64]</sub> *	<b>82.4</b> <sub>[0.37]</sub> *	<b>66.7</b> <sub>[0.66]</sub> *
<b>MSP w/o LF</b>	Emb.	93.0 <sub>[0.69]</sub>	83.2 <sub>[0.87]</sub> *	95.7 <sub>[0.54]</sub>	84.8 <sub>[0.22]</sub> *	98.7 <sub>[0.25]</sub>	81.8 <sub>[0.31]</sub>	58.3 <sub>[2.75]</sub>
	Enc.	96.2 <sub>[0.43]</sub> *	81.7 <sub>[0.73]</sub>	94.7 <sub>[0.30]</sub>	79.1 <sub>[0.60]</sub>	98.7 <sub>[0.14]</sub>	81.0 <sub>[0.20]</sub>	66.2 <sub>[0.54]</sub>
<b>MSP w/o MS</b>	Emb.	93.4 <sub>[0.52]</sub>	82.1 <sub>[1.00]</sub>	92.6 <sub>[0.52]</sub> *	83.7 <sub>[0.72]</sub> *	98.7 <sub>[0.19]</sub>	77.1 <sub>[0.38]</sub>	56.3 <sub>[1.72]</sub>
	Enc.	94.4 <sub>[0.86]</sub>	83.0 <sub>[0.41]</sub>	91.2 <sub>[0.81]</sub>	77.5 <sub>[1.27]</sub>	98.7 <sub>[0.27]</sub>	80.7 <sub>[0.27]</sub> *	66.5 <sub>[2.08]</sub> *

Table 4: The performance of different dialogue models on probing tasks for ablation experiments. Here we introduce three ablation settings: 1) **MSP**: The complete Multi-Source Probing method, 2) **MSP w/o LF**: MSP without the late fusion module, 3) **MSP w/o MS**: MSP without the multi-source attention mechanism and the late fusion module. Best results are marked in **bold**, and data that passed the significance test ( t-test, p-value < 0.05) are super-scripted with an asterisk \*.

Method		TREC	DNLI	MWOZ	SGD	SNIPS	SSA	Topic
<b>MSP</b>	Emb.	90.5 <sub>[0.27]</sub>	75.0 <sub>[0.64]</sub>	95.2 <sub>[0.27]</sub>	78.9 <sub>[0.80]</sub>	97.9 <sub>[0.11]</sub>	79.4 <sub>[0.28]</sub>	60.7 <sub>[1.19]</sub>
	Enc.	<b>91.2</b> <sub>[0.50]</sub> *	<b>76.2</b> <sub>[0.23]</sub> *	<b>95.4</b> <sub>[0.21]</sub> *	<b>80.2</b> <sub>[0.48]</sub> *	<b>98.3</b> <sub>[0.21]</sub> *	<b>80.7</b> <sub>[0.29]</sub> *	<b>63.1</b> <sub>[1.61]</sub> *
<b>MLP-Deep</b>	Embs.	83.7 <sub>[1.70]</sub> *	70.2 <sub>[0.48]</sub>	92.0 <sub>[0.36]</sub> *	69.2 <sub>[0.67]</sub> *	97.7 <sub>[0.20]</sub> *	77.3 <sub>[0.39]</sub>	56.4 <sub>[1.10]</sub>
	Enc.	80.7 <sub>[1.39]</sub>	69.3 <sub>[0.62]</sub>	89.5 <sub>[0.74]</sub>	67.5 <sub>[0.67]</sub>	96.9 <sub>[0.30]</sub>	78.4 <sub>[0.24]</sub> *	56.8 <sub>[2.72]</sub>

Table 5: The performance of the **MSP** and **MLP-Deep** methods on probing tasks. Best results are marked in **bold**, and data that passed the significance test ( t-test, p-value < 0.05) are super-scripted with an asterisk \*.

## A Discussion

### A.1 Eliminate the Effect of Parameter Scales

We have already found a positive correlation between parametric size and probe model performance earlier, so here we further explore the gap between MLP and MSP performance at the same parameter size to address the concern about the effect of the number of parameters on the probing results.

We extended the original two-layer MLP to a parameter scale consistent with the MSP, and the results are attached in Table 5. Although the parameter size of the MLP is increased to a level comparable to that of the MSP, the results are still sub-optimal because it does not operate in a manner consistent with the goals in the pretraining phase of the dialogue model. The supplementary experiment results demonstrate the effectiveness of the

MSP structure.

### A.2 Generalization on Pre-trained Language Models

We also evaluated the results of MSP over the state-of-the-art pre-trained language models. According to the results over BERT, BART, and T5 in Table 6, we could conclude that MSP still outperforms MLP even on the pre-trained language model.

Among the three models of comparable parameter size, Bart has the most outstanding ability for dialogue understanding. Bert performs optimally on the topic classification task, and T5 has very good performance on the task of intent detection, which is consistent with the characteristics and pre-training goals of the individual models themselves.

We found that Blenderbot-Small outperformed all of the three general pre-trained models in terms



Method		TREC	DNLI	MWOZ	SGD	SNIPS	SSA	Topic
<b>Bert MLP</b>	Embs.	87.8 <sup>*</sup> <sub>[0.55]</sub>	73.7 <sub>[0.77]</sub>	92.9 <sup>*</sup> <sub>[0.44]</sub>	73.0 <sup>*</sup> <sub>[0.68]</sub>	98.4 <sup>*</sup> <sub>[0.19]</sub>	78.2 <sub>[0.34]</sub>	46.9 <sub>[1.88]</sub>
	Enc.	85.7 <sub>[0.24]</sub>	81.8 <sup>*</sup> <sub>[0.45]</sub>	90.2 <sub>[0.39]</sub>	70.2 <sub>[0.34]</sub>	95.8 <sub>[0.19]</sub>	81.1 <sup>*</sup> <sub>[0.28]</sub>	59.9 <sup>*</sup> <sub>[1.08]</sub>
<b>Bert MSP</b>	Embs.	92.7 <sub>[0.36]</sub>	80.2 <sub>[0.54]</sub>	95.4 <sub>[0.18]</sub>	81.9 <sub>[0.37]</sub>	98.5 <sub>[0.56]</sub>	81.9 <sub>[0.47]</sub>	58.4 <sub>[1.54]</sub>
	Enc.	<b>94.9</b> <sup>*</sup> <sub>[0.23]</sub>	<b>86.7</b> <sup>*</sup> <sub>[0.59]</sub>	<b>95.8</b> <sup>*</sup> <sub>[0.42]</sub>	<b>83.2</b> <sup>*</sup> <sub>[0.44]</sub>	<b>98.6</b> <sup>*</sup> <sub>[0.68]</sub>	<b>82.3</b> <sup>*</sup> <sub>[0.97]</sub>	<b>69.7</b> <sup>*</sup> <sub>[1.12]</sub>
<b>Bart MLP</b>	Embs.	87.5 <sub>[0.57]</sub>	75.1 <sub>[0.95]</sub>	93.2 <sub>[0.50]</sub>	73.7 <sub>[0.60]</sub>	98.5 <sub>[0.12]</sub>	79.9 <sub>[0.12]</sub>	48.2 <sub>[2.13]</sub>
	Enc.	89.8 <sup>*</sup> <sub>[0.29]</sub>	85.9 <sup>*</sup> <sub>[0.36]</sub>	91.9 <sub>[0.24]</sub>	72.5 <sub>[0.39]</sub>	98.8 <sub>[0.05]</sub>	83.7 <sup>*</sup> <sub>[0.14]</sub>	56.0 <sup>*</sup> <sub>[0.75]</sub>
<b>Bart MSP</b>	Embs.	91.3 <sub>[0.33]</sub>	81.3 <sub>[0.87]</sub>	96.1 <sub>[0.25]</sub>	84.5 <sub>[0.79]</sub>	98.4 <sub>[0.23]</sub>	81.7 <sub>[0.21]</sub>	60.5 <sub>[1.74]</sub>
	Enc.	<b>95.6</b> <sup>*</sup> <sub>[0.29]</sub>	<b>87.6</b> <sup>*</sup> <sub>[0.46]</sub>	<b>96.3</b> <sup>*</sup> <sub>[0.41]</sub>	<b>87.2</b> <sup>*</sup> <sub>[0.07]</sub>	<b>99.0</b> <sup>*</sup> <sub>[0.18]</sub>	<b>84.9</b> <sup>*</sup> <sub>[0.22]</sub>	<b>68.1</b> <sup>*</sup> <sub>[0.76]</sub>
<b>T5 MLP</b>	Embs.	86.8 <sub>[1.54]</sub>	74.0 <sub>[1.07]</sub>	93.1 <sub>[0.54]</sub>	73.6 <sup>*</sup> <sub>[0.16]</sub>	97.2 <sub>[0.38]</sub>	78.5 <sub>[0.86]</sub>	44.5 <sub>[5.33]</sub>
	Enc.	89.5 <sup>*</sup> <sub>[0.26]</sub>	83.3 <sup>*</sup> <sub>[0.34]</sub>	93.1 <sub>[0.28]</sub>	72.4 <sub>[1.22]</sub>	98.1 <sub>[0.11]</sub>	82.1 <sup>*</sup> <sub>[0.20]</sub>	57.9 <sup>*</sup> <sub>[2.81]</sub>
<b>T5 MSP</b>	Embs.	91.7 <sub>[0.41]</sub>	80.8 <sub>[1.39]</sub>	95.6 <sub>[0.47]</sub>	83.2 <sub>[0.19]</sub>	98.6 <sub>[0.07]</sub>	81.4 <sub>[0.24]</sub>	59.1 <sub>[2.45]</sub>
	Enc.	<b>93.2</b> <sup>*</sup> <sub>[0.73]</sub>	<b>85.1</b> <sup>*</sup> <sub>[0.54]</sub>	<b>96.2</b> <sup>*</sup> <sub>[0.58]</sub>	<b>85.7</b> <sup>*</sup> <sub>[0.02]</sub>	<b>98.9</b> <sup>*</sup> <sub>[0.17]</sub>	<b>83.3</b> <sup>*</sup> <sub>[0.25]</sub>	<b>67.6</b> <sup>*</sup> <sub>[1.22]</sub>

Table 6: The performance of the MSP and MLP methods with the state-of-the-art pre-trained language models **BERT**, **BART**, and **T5** on several probing tasks. Best results are marked in **bold**, and data that passed the significance test ( t-test, p-value < 0.05) are super-scripted with an asterisk \*.

Method		TREC	DNLI	MWOZ	SGD	SNIPS	SSA	Topic
<b>MLP</b>	Rand.Emb.	71.8 <sub>[1.73]</sub>	60.4 <sub>[0.42]</sub>	87.2 <sub>[0.50]</sub>	60.5 <sub>[0.60]</sub>	96.6 <sub>[0.36]</sub>	70.6 <sub>[0.29]</sub>	39.8 <sub>[2.49]</sub>
	Orig.Emb.	83.1 <sub>[1.07]</sub>	69.9 <sup>*</sup> <sub>[0.79]</sub>	92.1 <sup>*</sup> <sub>[0.38]</sub>	69.9 <sup>*</sup> <sub>[0.55]</sub>	97.9 <sub>[0.24]</sub>	77.1 <sub>[0.17]</sub>	53.3 <sup>*</sup> <sub>[2.00]</sub>
<b>Linear</b>	Rand.Emb.	65.7 <sub>[1.70]</sub>	53.4 <sup>*</sup> <sub>[0.59]</sub>	84.9 <sub>[0.30]</sub>	55.4 <sub>[0.26]</sub>	95.5 <sub>[0.46]</sub>	69.2 <sup>*</sup> <sub>[0.25]</sub>	35.8 <sup>*</sup> <sub>[2.30]</sub>
	Orig.Emb.	66.0 <sub>[0.74]</sub>	51.2 <sub>[0.85]</sub>	84.2 <sub>[0.61]</sub>	63.8 <sub>[0.98]</sub>	95.5 <sub>[0.38]</sub>	62.3 <sub>[0.61]</sub>	31.2 <sub>[3.26]</sub>

Table 7: The performance of the **MLP** and **Linear-Probing** methods with original and random embeddings on several different probing tasks. Data that passed the significance test (t-test, p-value < 0.05) are super-scripted with an asterisk \*.

Model	Parameter	Perplexity
<b>Transformer</b>	37M	30.3
<b>BlenderBot<sub>SMALL</sub></b>	90M	10.4
<b>DialoGPT<sub>SMALL</sub></b>	117M	8.6

Table 8: Training results of dialogue models.

of accuracy on the MWOZ SGD and Topic tasks. While DialoGPT-Small performs best on the TREC and SNIPS tasks. Another point to note is that all three models included in the supplemental experiments have over 25% more parameters than the corresponding BlenderBot-Small DialoGPT.

### A.3 Non-linear Probing Finetuning

Nonlinear probing is widely adopted in previous works (Belinkov et al., 2017; Belinkov and Glass, 2017; Conneau et al., 2018). In fact, non-linear probing and linear probing are essentially similar in probing. Classifiers with even a shallow linear structure can still fit well on these probing tasks. To this end, we add an experiment on linear probing, in which the original/random embeddings of the dialogue model were connected to a single linear layer for linear regression. The results in Table 7 show

that the linear probing method can also achieve more than 95% accuracy on random embeddings in the SNIPS task. It also proves the previous conclusion that linear probing is not an exclusive probing skill and non-linear probing has stronger probing performance in many aspects. We also add the experimental results of MSP on Bert (See Table 6), where we could see that Bert does not outperform pretrained ODD models of comparable size on the task of probing for language understanding.

## B Dataset Examples

Examples of probing dataset are shown in Table 9.

## C Implementation Details

We implemented the above models with PyTorch (Paszke et al., 2017), OpenPrompt (Ding et al., 2021) and Huggingface Library (Wolf et al., 2019a). When implementing the Multi-Source Probing method upon DialoGPT, we introduced randomly initialized cross-attention parameters together with the decoder for fine-tuning since DialoGPT is not an encoder-decoder-based model. We utilized Soft Verbalizer (Hambarzumyan et al., 2021; Hu et al., 2022), where a continuous vector

Dataset	Type	Example	Class	Label
TREC	Question Classification	[User1]: What sprawling U.S. state boasts the most airports?	number, entity, location, ...	location
DialogueNLI	Natural Language Inference	[User1]: My locks are chestnut . [User1]: You are blonde .	entail, contradict, neutral	contradict
MultiWOZ	Intent Classification	[User1]: Can you find an expensive restaurant that serves Venetian food? [User2]: Sorry, looks like there aren't any Venetian restaurants that are expensive. Would you like something else? [User1]: Yes, I would be interested in one that serves Chinese food. Where would you recommend?	restaurant-inform, taxi-request, general-thank, ...	restaurant-inform
Schema-Guided	Intent Classification	[User1]: Can you show me attractions I can visit? [User2]: Where do you want me to search. [User1]: In Toronto please. [User2]: I found 10, how about 3D Toronto Sign, it's a tourist attraction.	find-attractions, reserve-flight, get-ride, ...	find-attractions
SNIPS	Intent Classification	[User1]: I need another artist in the New Indie Mix playlist.	search-screening, add-to-playlist, get-weather, ...	add-to-playlist
ScenrioSA	Sentiment Classification	[User1]: Have you met the new intern? [User2]: Yes. She's very enthusiastic. [User1]: I know. I don't trust her. [User2]: Why? She is just ambitious. [User1]: I think she wants my job.	positive, negative, neutral	negative
DailyDialog Topic	Topic Classification	[User1]: Happy birthday , Jim ! Here is a present for you. [User2]: Oh , great ! I love it! [User1]: I'm very glad to hear that . [User2]: Come here, let me introduce some friends to you.	ordinary life, work, school, tourism, politics, relationship, ...	relationship

Table 9: Examples from probing tasks.

is designed for each class label, to generate the probability distribution for class label space by calculating the dot product between the output of the language model and the class vector. The class vectors are initialized with the pretrained token embeddings and will be fine-tuned through training.

All models in this paper are optimized through ADAM (Kingma and Ba, 2014) with learning rate and dropout rate optimized through grid search. The number of soft tokens is empirically set to 12 in our Multi-Source Probing method. The learning rate is searched within the range of  $\{5 \times 10^{-5}, 1 \times 10^{-4}, 2 \times 10^{-4}, 3 \times 10^{-4}, 5 \times 10^{-4}\}$  and dropout rate within  $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ . The batch size for Transformer, BlenderBot, and DialoGPT are 128, 64, and 16 respectively. Accuracy and standard deviation data in this paper are calculated from the results of 5 replicate experiments. We conducted our probing experiments on the NVIDIA V100 Tensor Cores, the average run-time for each probing task is about 5 hours.