Text is no more Enough! A Benchmark for Profile-based Spoken Language Understanding

Xiao Xu[1*], Libo Qin[1*], Kaiji Chen[2], Guoxing Wu[2], Linlin Li[2], Wanxiang Che[1†], Ting Liu[1]
[1]Research Center for Social Computing and Information Retrieval
[1]Harbin Institute of Technology, China
[2]Huawei Technologies Co., Ltd.
{xxu,lbqin,car,tliu}@ir.hit.edu.cn, {chenkaiji,wuguoxing1,lynn.lilinlin}@huawei.com

Abstract

Current researches on spoken language understanding (SLU) heavily are limited to a simple setting: the plain text-based SLU that takes the user utterance as input and generates its corresponding semantic frames (e.g., intent and slots). Unfortunately, such a simple setting may fail to work in complex real-world scenarios when an utterance is semantically ambiguous, which cannot be achieved by the text-based SLU models. In this paper, we first introduce a new and important task, **Pro**file-based **S**poken **L**anguage **U**nderstanding (PROSLU), which requires the model that not only relies on the plain text but also the supporting profile information to predict the correct intents and slots. To this end, we further introduce a large-scale human-annotated Chinese dataset with over 5K utterances and their corresponding supporting profile information (*Knowledge Graph* (KG), *User Profile* (UP), *Context Awareness* (CA)). In addition, we evaluate several state-of-the-art baseline models and explore a multi-level knowledge adapter to effectively incorporate profile information. Experimental results reveal that all existing text-based SLU models fail to work when the utterances are semantically ambiguous and our proposed framework can effectively fuse the supporting information for *sentence-level* intent detection and *token-level* slot filling. Finally, we summarize key challenges and provide new points for future directions, which hopes to facilitate the research.

1 Introduction

Spoken Language Understanding (SLU) (Young et al. 2013; Qin et al. 2021d) is a core component in task-oriented dialogue systems, aiming to extract intent and semantic constituents from the natural language utterances (Tur and De Mori 2011). It consists of two typical subtasks: intent detection and slot filling to map the user input utterance into an overall intent and a slot label sequence.

With the help of pre-trained models, recent work has achieved remarkable success on the SLU system. Qin et al. (2021d) surveys that performance improvement on traditional SLU is relatively already saturated because the neural joint model has achieved over 96% and 99% on slot filling and intent detection on the ATIS dataset (Hemphill, Godfrey, and Doddington 1990). Though achieving good performance, the current researches on SLU mainly focus on a simple scenario: the plain text-based setting.

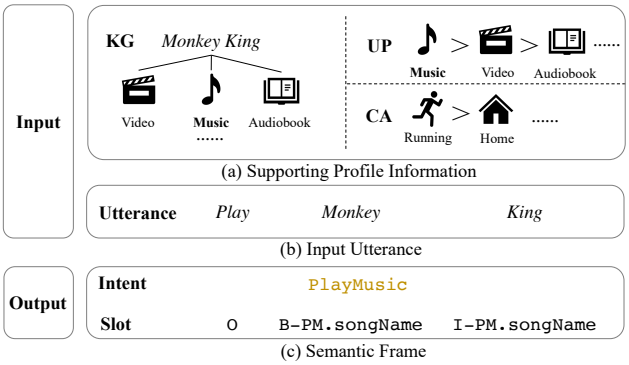

Figure 1: An example in PROSLU. Semantic frame denotes the intent and slots of the input utterance. The ">" implies that the probability of the former is greater than the latter.

More specifically, traditional SLU systems are based on the assumption that simply relying on plain text can capture intent and slots correctly. Unfortunately, such an assumption may not be achieved in real-world scenarios when the user utterance is semantically ambiguous. For example, as shown in Figure 1(b)&(c), when a user asks the agent (e.g., Apple Siri) for the query "*Play Monkey King*", simply relying on the text is not enough for extracting correct semantic frame results, since "*Monkey King*" could indicate a "`rock song`" or an "`eponymous Chinese TV cartoon`". *Therefore, we argue that the existing text-based SLU is not enough for the complex setting in real-world scenarios when the utterance is semantically ambiguous.* In this paper, we assume that the profile information of the user can help to solve the issue where the corresponding profile information can be used as supplementary knowledge to alleviate the ambiguity of utterance. As illustrated in Figure 1(a), if a user is running and prefers music to video, *Monkey King* is more likely to be music than video. Unfortunately, none of the work considers the profile-based SLU in real-world scenarios. One of the key reasons for hindering the progress is the lacking of public benchmarks.

*These authors contributed equally.
†Email corresponding.

Input	
Utterance	*Play Monkey King*
KG	`Mention` "*Monkey King*": {music, video and audiobook}, ...
UP	`Preference` for [music, video, audiobook]: [0.5, 0.3, 0.2], ...
CA	`Movement State`: Running, `Geographic Location`: Home, ...
Output	
Intent	`PlayMusic`
Slot	`O B-PlayMusic.songName I-PlayMusic.songName`

Table 1: A simplified example from the PROSLU dataset.

In the paper, to bridge the research gap, we propose a new and important task, **Pro**file-based **S**poken **L**anguage **U**nderstanding (PROSLU), which requires a model not only depends on the text but also on the given supporting profile information. We further introduce a Chinese human-annotated dataset, with over 5K utterances annotated with intent and slots, and corresponding supporting profile information. In total, we provide three types of supporting profile information: (1) *Knowledge Graph* (KG) consists of entities with rich attributes, (2) *User Profile* (UP) is composed of user settings and information, (3) *Context Awareness* (CA) is user state and environmental information.

To establish baselines on PROSLU, we evaluate several state-of-the-art models. The experimental results reveal that all models fail to work (up to 50% in overall accuracy metric) on PROSLU. In addition, we propose a *multi-level* knowledge adapter to equip the existing SLU models with the ability to incorporate profile information, which has the following advantages: (1) it achieves a fine-grained knowledge injection for both *sentence-level* intent detection and *token-level* slot filling; (2) it can be used as a plugin and easily be compatible with the existing state-of-the-art SLU models.

Contributions of this work are concluded as:

- We systematically analyze the state-of-the-art SLU models and observe that existing models fail to work in real-world scenarios, which shed a light for future research.
- We propose a new and important task named PROSLU. In addition, we introduce a Chinese human-annotated dataset, hoping it would push forward further research. To our knowledge, we are the first to explore PROSLU.
- We establish various baselines and conduct qualitative analysis for PROSLU. Besides, we explore a *multi-level* adapter to effectively inject the profile information.

We hope the task and datasets will invite more research on PROSLU. All datasets and codes used in this paper are publicly available at https://github.com/LooperXX/ProSLU.

2 Problem Definition

In this section, we define the supporting profile information, profile-based intent detection and slot filling. A simplified example is given in Table 1 and the complete example can be found in the Appendix A.4.

2.1 Supporting Profile Information

We introduce three types of supporting profile information including *Knowledge Graph*, *User Profile* and *Context Awareness*, which are used to help the model to alleviate the ambiguity in the utterances.

Knowledge Graph The first type of profile information is the **K**nowledge **G**raph (KG), which contains large amounts of interlinked entities and their corresponding rich attributes[1]. Depending on the context, an ambiguous mention refers to some different entities of the same (or similar) name but different entity types, as the ambiguous mention (or their shared name) tends to be polysemous (i.e., have multiple meanings). Take Table 1 for example, KG information provides background knowledge for the ambiguous mention *Monkey King* (e.g., it can be an entity of music, video or audiobook). Following Chen et al. (2020), we represent each entity and its attributes in KG as a long text sequence, which is composed of key-value pairs (e.g., "`subject`: *Monkey King*, `type`: CreativeWork").

User Profile The second type of profile information is the **U**ser **P**rofile (UP), which is a collection of settings and information (items) associated with the user. Each item in UP consists of a non-negative float array that sums to 1. As shown in Table 1, the user "preferences for music, video, and audiobook: $[0.5, 0.3, 0.2]$" is an item in UP information, which can help the model to judge that the user prefers listening to music rather than watching videos. We concat all the items in UP and directly flatten them to a single feature vector $\mathbf{x}_{\text{UP}} \in \mathbb{R}^u$ (u is the UP feature dimension). For example, the user preferences for music, video, and audiobook are $[0.5, 0.3, 0.2]$ and the user transportation preferences for subway, bus and driving are $[0.4, 0.1, 0.5]$, we could get $[0.5, 0.3, 0.2, 0.4, 0.1, 0.5]$.

Context Awareness The third type of information is the **C**ontext **A**wareness (CA) that denotes the user state and environmental information, including the user's movement state, posture, geographic location, etc. As shown in Table 1, a user who is running is more likely to play music than video. The form of each item in CA is similar to UP, e.g., the movement state can be walking, running, or stationary, and [0,1,0] indicates that the movement state is running Similarly, we get the flatten feature vector $\mathbf{x}_{\text{CA}} \in \mathbb{R}^c$ (c is the CA feature dimension).

2.2 Profile-based Intent Detection and Slot Filling

Unlike the traditional SLU task, PROSLU requires the model to predict results not only rely on the input utterance, but also the corresponding supporting profile information.

Specifically, given an input word sequence $\mathbf{x} = (x_1, \ldots, x_T)$ (T is the number of words) and its corresponding supporting profile information, profile-based intent detection can be seen as a sentence classification problem to decide the intent label o^{I} while profile-based slot filling is a sequence labeling task to generate a slot sequence $\mathbf{o}^{\text{S}} = (o_1^{\text{S}}, \ldots, o_T^{\text{S}})$. Formally, the PROSLU task can be defined as:

$$(o^{\text{I}}, \mathbf{o}^{\text{S}}) = f(X, \text{KG}, \text{UP}, \text{CA}), \tag{1}$$

where f denotes the trained model.

[1]We use open-source encyclopedia knowledge graphs like CN-DBpedia, OwnThink, etc.

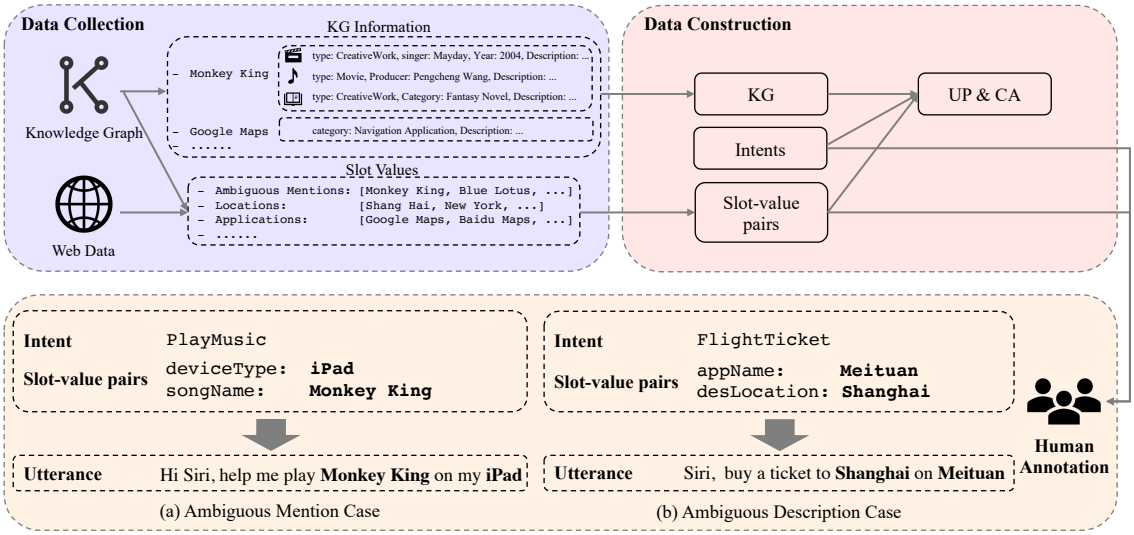

Figure 2: The overall workflow of our data collection, data construction and human annotation.

3 Dataset

In this section, we describe the collection and annotation process of the PROSLU dataset. In PROSLU, each utterance is semantically ambiguous, which requires the model to leverage the supporting profile information.

3.1 Ambiguity Definition

By manually collecting semantically ambiguous samples from the error cases in real-world systems, we found that there are two main sources of ambiguity in user utterances in real-world scenarios, including ambiguous mentions and ambiguous descriptions.

Ambiguous Mentions It indicates that the presence of ambiguous mentions in the user utterance introduces lexical ambiguity and ultimately leads to semantic ambiguity in the utterance. For example, the ambiguous mention *Monkey King* can represent different entities such as a rock song (sung by the Mayday band), a biographical novel, or an eponymous Chinese TV cartoon.

Ambiguous Descriptions It indicates that the ambiguity is caused by the ambiguous semantic understanding of utterance rather than the ambiguous entities. Take the user utterance "I want to buy a ticket to Shanghai" as an example, it's hard to capture the correct intent simply depending on the utterance, because the intent of utterance could be to book a train ticket, a plane ticket, or a coach ticket.

3.2 Data Design

Intent and Slot Design We design multiple ambiguous intent groups based on real-world scenarios and open-source SLU datasets. For example, {`PlayMusic`, `PlayVideo`, `PlayAudioBook`} is an ambiguous intent group where each intent is ambiguous with each other ("Play Monkey King" can be any intent in this ambiguous intent group). Slot labels are collected directly from the slot label sets corresponding to the intents.

Supporting Profile Information Design For KG information, we collect it directly from the open-source knowledge graphs. For UP and CA information, based on the UP and CA schemas in real scenarios, we carefully pick out the items that can help disambiguate the above intents and slot labels, and integrate them to form the final UP and CA items.

3.3 Data Collection

We first collect values for different slots based on the crawled public web data. Then we collect ambiguous mentions from the knowledge graph. The entities could share the same mentions but have different entity types in the knowledge graph. For example, the song entity *Monkey King* and the Chinese TV cartoon entity *Monkey King* has the same mention (name) but different entity types. Therefore, we are able to collect numerous ambiguous mentions.

3.4 Data Construction

Based on the first two steps, we design the data generation process separately for these two ambiguity cases defined in Section 3.1.

Ambiguous Description Case For each sample, we first randomly select an intent and some slots in its corresponding slot label sets. Next, we fill these slots by randomly selecting slot values from the collected slot values. In addition, when some slot values are entities in the knowledge graph, we extract KG information for them. Finally, we design the heuristic rules to generate valid UP and CA information for the corresponding intent. For example, when the selected intent is `SearchDriveRoute` (search for driving routes between two places), the "`Has Car`" item in UP information is more likely to be "true" and the "`Movement State`" in

#Utterances	5,249
#Utterances in Train Set	4,196
#Utterances in Valid Set	522
#Utterances in Test Set	531
#Avg. Words per Utterance	23.64
#Intents	14
#Slots	49
#UP	4
#CA	4
#KG entities	7,466
#Avg. KG Entity	2.77
#Avg. Words per Entity	272.63

Table 2: Data statistics of PROSLU dataset. #Avg. KG Entity denotes the average number of entity per data sample.

CA information may be less likely to be "on the aircraft"[2].

Ambiguous Mention Case To bring lexical ambiguity into the utterance, there should exist ambiguous mentions in the slot values of the utterance. Thus, slightly different from the former case, after randomly selecting the intent and slots, the ambiguous mention should be randomly selected but satisfies the selected intent[3]. Then we generate slot-value pairs and obtain the KG information of different entities corresponding to the selected mention from the knowledge graph data. Based on the selected intent and the entities in the KG information, we design the hard-coded heuristics to randomly generate valid UP and CA information. For example, for the `PlayVideo` intent, the movement state in CA information is less likely to be "running", and if the entity types of entities in the KG information are {music, video and audiobook}, the user preferences for video tend to greater than music and audiobook.

3.5 Human Annotation

After data collection and construction, the annotators only need to manually write the ambiguous utterances in conjunction with the given intent and slot-value pairs.

We hire an annotation team to check the generated data in each given sample and to annotate the utterances. More importantly, the utterances annotated by the annotators must be reasonable and logical but semantically ambiguous. The sample with unreasonable generated data will be removed. Figure 2 gives an illustration of the overall workflow.

3.6 Quality Control

To ensure quality, each sample is annotated by three experts and the annotation process lasts for nearly two months. In practice, we randomly divide all the completed annotated samples into 10 groups and select 50 sentences from each group for testing, and if more than 5 sentences are regarded as incorrectly annotated, the whole group would be re-annotated. Finally, we obtain 5,249 samples, where the ratio of description ambiguity vs. mention ambiguity case in the dataset is nearly 1:2. Table 2 summarizes the detailed statistics of the PROSLU dataset.

[2]Users who do have a car are usually more likely to ask for driving routes, and users who do not have an Internet connection on the aircraft are usually less likely to try to search.

[3]For example, the selected mention must have a music entity to satisfy the `PlayMusic` intent.

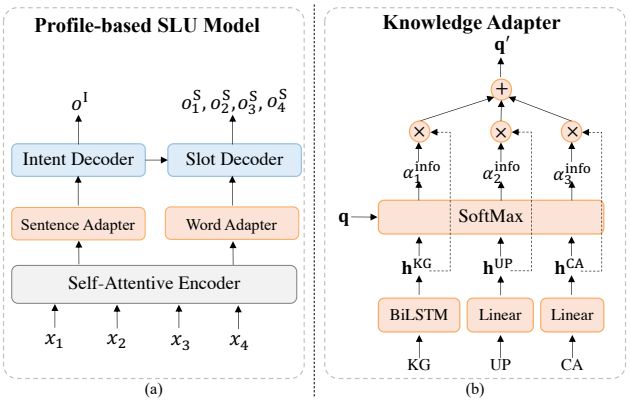

Figure 3: The illustration of Profile-based SLU model (a), which consists of a general SLU model in Section 4.1 and a knowledge adapter (b) in Section 4.3.

4 Approach

In this section, we first introduce the general SLU model and then describe the proposed multi-level knowledge adapter , which can be used for *sentence-level* intent detection and *word-level* slot filling, respectively [4].

4.1 General SLU Model

The general SLU model consists of a shared encoder, an intent detection decoder, and a slot filling decoder.

Shared Encoder The shared encoder reads the input utterance $\mathbf{x} = \{x_1, x_2, .., x_T\}$ (T is the number of tokens in the input utterance) to generate the shared encoding representation $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_T\} = \text{Encoder}\,(x_1, x_2, .., x_T)$.

Intent Detection Decoder Based on the shared encoding representation $\mathbf{E}$, a sentence representation $\mathbf{g}$ can be generated (e.g., the sentence self-attention mechanism (Zhong, Xiong, and Socher 2018)) for intent detection:

$$\mathbf{y}^{\mathrm{I}} = \text{softmax}\,(\mathbf{W}_{\mathrm{I}}\,\mathbf{g})\,, \quad (2)$$

$$o^{\mathrm{I}} = \arg\max(\mathbf{y}^{\mathrm{I}}), \quad (3)$$

where $\mathbf{W}_{\mathrm{I}}$ are trainable parameters.

Slot Filling Decoder The unidirectional LSTM is used as the slot filling decoder. At each decoding step t, we adopt the intent-guided mechanism (Qin et al. 2019) and the decoder state $\mathbf{h}_t^{\mathrm{S}}$ is a function of the previous decoder state $\mathbf{h}_{t-1}^{\mathrm{S}}$, the aligned encoder hidden state $\mathbf{e}_t$, the embedding of predicted intent and previously emitted slot.

Finally, $\mathbf{h}_t^{\mathrm{S}}$ is used for slot filling:

$$\mathbf{y}_t^{\mathrm{S}} = \text{softmax}\left(\mathbf{W}_{\mathrm{S}}\,\mathbf{h}_t^{\mathrm{S}}\right), \quad (4)$$

$$o_t^{\mathrm{S}} = \arg\max(\mathbf{y}_t^{\mathrm{S}}), \quad (5)$$

[4]The detailed description and training objectives of the general SLU model can be found in the Appendix A.3.

where o_t^{S} is the slot label of the t-th word in the utterance and $\mathbf{W}_{\mathrm{S}}$ are trainable parameters.

4.2 Supporting Information Representations

KG Representation KG information of each entity is a concatenated sequence of key-value pairs. For KG information with only one entity, e.g., $\mathbf{x}^{\mathrm{KG}} = \{x_1^{\mathrm{KG}}, x_2^{\mathrm{KG}}, \ldots, x_L^{\mathrm{KG}}\}$ (L is the number of words in the KG sequence), we use BiLSTM to obtain the KG encoding representations $\mathbf{H}^{\mathrm{KG}} = \{\mathbf{h}_1^{\mathrm{KG}}, \mathbf{h}_2^{\mathrm{KG}}, \ldots, \mathbf{h}_L^{\mathrm{KG}}\} \in \mathbb{R}^{L \times d_i}$ by applying $\mathbf{h}_t^{\mathrm{KG}} = \mathrm{BiLSTM}\left(\phi^{\mathrm{KG}}\left(x_t^{\mathrm{KG}}\right), \mathbf{h}_{t-1}^{\mathrm{KG}}\right)$, where d_i is the information embedding dimension. We directly use the last hidden state $\mathbf{h}_L^{\mathrm{KG}}$ as the KG representation $\mathbf{h}^{\mathrm{KG}}$. For KG information consisting of sequences of multiple entities, we perform average pooling for the last hidden state of each sequence as the overall aggregated KG representation $\mathbf{h}^{\mathrm{KG}}$.

UP and CA Representation UP and CA representations can be obtained by using linear projection, using $\mathbf{h}^{\mathrm{UP}} = \mathbf{W}_{\mathrm{UP}}^{\top}\, \mathbf{x}_{\mathrm{UP}}$ and $\mathbf{h}^{\mathrm{CA}} = \mathbf{W}_{\mathrm{CA}}^{\top}\, \mathbf{x}_{\mathrm{CA}}$, respectively, where $\mathbf{W}_{\mathrm{UP}} \in \mathbb{R}^{u \times d_i}$ and $\mathbf{W}_{\mathrm{CA}} \in \mathbb{R}^{c \times d_i}$ are trainable parameters.

4.3 Multi-level Knowledge Adapter

Knowledge Adapter The core challenge of PROSLU is how to effectively incorporate the supporting information. We explore a knowledge adapter to address this challenge, which can be used as a plugin without changing the original SLU structure. Inspired by Srinivasan et al. (2020), we adopt the hierarchical attention fusion mechanism (Luong, Pham, and Manning 2015; Libovický and Helcl 2017) as the knowledge adapter, which has the advantage of dynamically considering relevant supporting information for different words. Specifically, given the query vector $\mathbf{q}$ and the corresponding supporting information $\mathbf{H}^{\mathrm{Info}} = [\mathbf{h}^{\mathrm{KG}}; \mathbf{h}^{\mathrm{UP}}; \mathbf{h}^{\mathrm{CA}}] \in \mathbb{R}^{3 \times d_i}$, we obtain the updated representation $\mathbf{q}'$ = Knowledge-Adapter($\mathbf{q}, \mathbf{H}^{\mathrm{info}}$) by weighted summing the representation from all the supporting information:

$$\alpha_i^{\mathrm{info}} = \frac{\exp\left(\mathbf{q}\,\mathbf{W}\,\mathbf{h}_i^{\mathrm{info}}\right)}{\sum_{k=1}^{3} \exp\left(\mathbf{q}\,\mathbf{W}\,\mathbf{h}_k^{\mathrm{info}}\right)}, \tag{6}$$

$$\mathbf{q}' = \sum_{i=1}^{3} \alpha_i^{\mathrm{info}} \mathbf{h}_i^{\mathrm{info}}, \tag{7}$$

where $\mathbf{W}$ are trainable parameters and $\{\mathbf{h}_1^{\mathrm{Info}}, \mathbf{h}_2^{\mathrm{Info}}, \mathbf{h}_3^{\mathrm{Info}}\}$ denotes $\{\mathbf{h}^{\mathrm{KG}}, \mathbf{h}^{\mathrm{UP}}, \mathbf{h}^{\mathrm{CA}}\}$ respectively.

Sentence-level Knowledge Adapter for Intent Detection We perform a sentence-level knowledge adapter for sentence-level intent detection, where we use sentence representation $\mathbf{g}$ as query to obtain the hierarchical fused information $\mathbf{s}^{\mathrm{info}}$, using $\mathbf{s}^{\mathrm{info}}$ = Knowledge-Adapter($\mathbf{g}, \mathbf{H}^{\mathrm{info}}$), which is used for augmenting intent detection:

$$\mathbf{y}^{\mathrm{I}} = \mathrm{softmax}\left(\mathbf{W}_{\mathrm{I}}\left(\mathbf{g} \oplus \mathbf{s}^{\mathrm{info}}\right)\right). \tag{8}$$

Word-level Knowledge Adapter for Slot Filling Since slot filling is a word-level sequence labeling task, we apply a word-level knowledge adapter to inject different relevant knowledge for each word.

Specifically, we use the self-attentive encoding $\mathbf{e}_t$ at the t-th timestep as query vector to fuse supporting information using $\mathbf{s}_t^{\mathrm{info}}$ = Knowledge-Adapter($\mathbf{e}_t, \mathbf{H}^{\mathrm{info}}$). Similarly, $\mathbf{s}_t^{\mathrm{info}}$ is used to enhance word-level representation in slot filling decoder:

$$\mathbf{h}_t^{\mathrm{S}} = \mathrm{LSTM}\left(\mathbf{s}_t \oplus \mathbf{s}_t^{\mathrm{info}}, \mathbf{h}_{t-1}^{\mathrm{S}}\right) \tag{9}$$

$$\mathbf{y}_t^{\mathrm{S}} = \mathrm{softmax}\left(\mathbf{W}_{\mathrm{S}} \mathbf{h}_t^{\mathrm{S}}\right). \tag{10}$$

where $\mathbf{s}_t$ is the concatenation of the aligned encoder hidden state, intent embedding, and the previous slot embedding.

5 Experiments

5.1 Experimental Settings

The self-attentive encoder hidden units are 256 in all datasets. ℓ_2 regularization is 1×10^{-6} and the dropout rate is 0.4 for reducing overfitting. We use Adam (Kingma and Ba 2014) to optimize the parameters in our model and adopt the suggested hyper-parameters for optimization. For all the experiments, we select the model which works best on the dev set and then evaluate it on the test set. All experiments are performed on the GPU Tesla V100.

5.2 Baselines

We experiment the existing state-of-the-art non pre-trained SLU models on the PROSLU dataset: 1) `Attention BiRNN`. Liu and Lane (2016) proposes an alignment-based RNN with the attention mechanism, which implicitly models the relationship between slot and intent. 2) `Slot-Gated Atten`. Goo et al. (2018) proposes a slot-gated joint model to explicitly model the correlation between slot filling and intent detection. 3) `Bi-Model`. Wang, Shen, and Jin (2018) proposes the Bi-model to study the cross-impact between the intent detection and slot filling. 4) `SF-ID Network`. E et al. (2019) proposes an SF-ID network to construct direct connections for the slot filling and intent detection. 5) `Stack-Propagation` Qin et al. (2019) adopts a joint model with Stack-Propagation to capture the intent semantic knowledge. 6) `DCA-Net` Qin et al. (2021b) proposes a co-interactive transformer to consider the cross-impact between the two tasks. We also explore the existing state-of-the-art multi-intent models in the Appendix A.1.

To investigate the impact of pre-trained models in our PROSLU dataset, based on the general SLU model. we adopt the pre-trained models `BERT` (Devlin et al. 2019), `XLNet` (Yang et al. 2019), `RoBERTa` (Liu et al. 2019), `ELECTRA` (Clark et al. 2020) as the shared encoder to get the pre-trained-based SLU models.

5.3 Analysis on Baselines without Profile Information

Following Goo et al. (2018) and Qin et al. (2019), we evaluate the performance of slot filling using F1 score, intent detection using accuracy, the sentence-level semantic frame parsing using overall accuracy which represents all metrics are right in an utterance.

Model	w/o Profile			w/ Profile		
	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
Non Pre-trained SLU Models						
Attention BiRNN (Liu and Lane 2016)	35.42	43.86	31.38	79.68	82.67	75.89
Slot-Gated (Goo et al. 2018)	36.53	41.24	32.02	73.92	83.24	68.55
Bi-Model (Wang, Shen, and Jin 2018)	37.37	44.63	32.58	77.08	80.04	71.94
SF-ID (E et al. 2019)	39.63	42.37	30.89	72.36	83.99	65.35
Stack-Propagation (Qin et al. 2019)	39.29	39.74	36.35	81.08	83.99	78.91
DCA-Net (Qin et al. 2021b)	35.02	40.49	24.67	81.79	83.80	73.45
Pre-trained-based SLU Models						
BERT (Devlin et al. 2019)	44.80	45.76	42.18	82.51	84.56	80.98
XLNet (Yang et al. 2019)	**46.61**	**48.40**	**44.07**	83.39	**85.88**	81.73
RoBERTa (Liu et al. 2019)	45.92	47.83	43.13	82.90	85.31	81.17
ELECTRA (Clark et al. 2020)	46.48	47.46	42.56	**85.03**	85.12	**82.11**

Table 3: Slot Filling and Intent Detection results on the PROSLU dataset.

Non Pre-Trained SLU Models Performance We conduct experiments on PROSLU to observe the performance of the non pre-trained SLU models without supporting profile information. The results are shown in Table 3. We observe that all baseline models significantly drop a lot compared with ATIS and SNIPS dataset on all three metrics. For example, the baseline model Stack-Propagation (Qin et al. 2019) achieved 86.5% and 86.9% on overall accuracy on ATIS and SNIPS but only obtain 36.35% on the PROSLU dataset. This indicates that the existing models fail to work when the utterances are semantically ambiguous.

Pre-Trained-based SLU Models Performance In this section, we further conduct experiments with the pre-trained-based SLU models on PROSLU to observe its performance without supporting profile information. The same trend is observed in Table 3. The overall accuracy of all pre-trained-based SLU models (w/o Profile) is still less than 45%, which indicates that simply using pre-trained models does not alleviate the situation.

Comparison Between Non Pre-Trained Models and Pre-Trained Models Comparing with non pre-trained models, we can see that all the pre-trained-based SLU models are better than the non pre-trained SLU models, which can bring 5% to 7% improvement. We attribute this to the fact that the pre-trained models learn general semantic knowledge in the pre-training stage, hence it can provide rich semantic features that can help to ease the ambiguity in PROSLU.

5.4 Analysis on Baselines with Profile Information

Table 3 (w/ Profile column) shows the performance of all models with supporting profile information on the PROSLU dataset. It can be seen that the performance of all models improve significantly by a large margin based on our multi-level knowledge adapter to incorporate supporting profile information. All the three metrics improve by about 30% to 40%, which indicates the supporting profile information can help alleviate the ambiguity in the ambiguous utterances. It further proves the significance and importance of our PROSLU task for real-world scenarios.

Model	Slot (F1)	Intent (Acc)	Overall (Acc)
ELECTRA	**85.03**	**85.12**	**82.11**
w/o Sentence-level Adapter	77.32	48.78	43.88
w/o Word-level Adapter	79.99	81.36	78.15
w/o Multi-level Adapter	46.48	47.46	42.56

Table 4: Ablation Study of Multi-level Knowledge Adapter.

5.5 Ablation Study of Multi-level Knowledge Adapter

To explore the effectiveness of multi-level knowledge adapter for PROSLU task, we perform the ablation study on the best model, ELECTRA-based SLU model (w/ Profile), in Table 4.

Effect of Sentence-level Adapter We first experiment by only adopting a word-level slot adapter and the results are shown in Table 4 (w/o Sentence-level Adapter Row), we observe a significant decrease in all three metrics, 7.71%, 36.34%, 38.23%, respectively, and performance degradation is most obvious on intent detection task. This is because the sentence-level intent adapter can effectively help the intent detection task to identify the correct intent and transfer the correct guidance knowledge for the slot filling task through explicitly interacting between two tasks.

Effect of Word-level Adapter We remove the word-level slot adapter and only adopt the sentence-level intent adapter, which means there is no direct supporting information is injected into the slot filling decoder. The results are shown in Table 4 (w/o Word-level Adapter Row). We observe that our framework drops 5.04% in slot filling task, which indicates that the word-level adapter can effectively inject profile knowledge for word-level slot filling task. An interesting observation is that the performance decrease of three metrics is slightly lower compared to w/o Sentence-level Adapter. We assume that although we remove the world-level adapter, the remaining sentence-level adapter can still help train a good intent detector, which can be used to guide the slot filling task with the intent-guided mechanism we adopted.

Utterance	*Open my iPad and search for Answer , sung by Joey Yung*
Supporting	**KG**: Mention "*Answer*": **7** entities of{music or audiobook}
Slot	Predict: O O deviceType O O O songName O artist artist Real: O O deviceType O O O O O artist artist
Utterance	*Play Martial Universe by Hu Li on my iPad*
Supporting	**KG**: Mention "*Martial Universe*": **3** entities of {video or audiobook}, ... **UP**: Preference for [music, video, audiobook]: [0.4, 0.2, 0.4], ... **CA**: Movement State: Walking, Geographic Location: Home, ...
Intent	Predict: PlayVideo Real: PlayAudioBook

Table 5: Error examples in ELECTRA-based SLU model(w/ Profile). Some information in KG, UP and CA are omitted for brevity.

Effect of Multi-level Adapter We remove the proposed multi-level adapter and directly conduct experiments with the ELECTRA-based SLU model (w/o Profile). As shown in Table 4 (w/o Multi-level Adapter Row), we observe a more significant decrease in all three metrics, 38.55%, 37.66%, 39.55%, respectively. This further demonstrates that our multi-level adapter can effectively incorporate supporting profile information into the intent detection task and slot filling task, achieving fine-grained knowledge transfer to effectively cope with ambiguous sentences in real scenarios.

5.6 Error Analysis

In this section, we empirically provide error samples of two different types generated from ELECTRA-based SLU model(w/ Profile).

KG Representation As shown in the first block of Table 5, when the KG information shows the "*Answer*" can be the music or audiobook entity, the model predicts "*Answer*" as "O" incorrectly. We attribute it to the fact that we simply represent each entity by flatting its KG information into a sequence and perform average pooling to obtain the overall aggregated KG representation, which would not effectively and correctly represent the KG information.

Supporting Profile Information Fusion Take the second block in Table 5 as an example, although the supporting profile information shows that the user prefers listening to *audiobook* rather than *watching videos*, the model predicts the intent as PlayVideo incorrectly, which means the knowledge fusion between the three types of supporting profile information needs to be more accurate and effective.

5.7 Challenges

Based on above analysis, we summarize the current challenges for our PROSLU dataset.

Incorporation of KG Information In this paper, we follow Chen et al. (2020) to represent KG information as long text sequences composed of key-value pairs, which poses a huge challenge for the KG encoder. It is an interesting question to investigate how to encode long KG sequences effectively. In addition, it is also possible to train the representation of each entity directly on the knowledge graph through knowledge graph embedding methods (Bordes et al. 2013).

Effectiveness of Fusion Approaches We follow Libovický and Helcl (2017) to adopt the hierarchical attention fusion mechanism to fuse the supporting profile information. Many representation fusion approaches exist in the machine learning research area (Zadeh et al. 2017; Liu et al. 2018; Tsai et al. 2019). It will be challenging and rewarding to explore the effectiveness of these approaches on PROSLU.

Expansion of Supporting Profile Information In this paper, we investigate three types of supporting profile information, which is common in real-world scenarios. To better solve PROSLU task and alleviate ambiguity from user utterances, more types of supporting profile information can be expanded in future research.

6 Related Work

Dominant SLU systems adopt the joint model to jointly consider the correlation between intent detection and slot filling (Qin et al. 2021d). Zhang and Wang (2016) and Hakkani-Tür et al. (2016) propose the multi-task framework to jointly model the correlation between the two tasks. Goo et al. (2018); Qin et al. (2019) and Teng et al. (2021) explicitly incorporate intent information for guiding the slot filling. Another series of work (Li, Li, and Qi 2018; E et al. 2019; Qin et al. 2021b) consider the bidirectional connection between the two tasks. Zhu, Cao, and Yu (2020) also considers the semi-supervised NLU setting. However, their work mainly focus on the plain text-based SLU. In contrast, we mainly consider the ambiguous setting and propose a new and important task, PROSLU which requires a model to predict the intent and slots correctly given text and its supporting profile information.

The ambiguous problem has attracted increasing attention in dialogue direction. Bhargava et al. (2013); Xu and Sarikaya (2014); Chen et al. (2015, 2016); Su, Yuan, and Chen (2018); Qin et al. (2021a) have shown leveraging contextual information can handle the ambiguous problem in SLU direction. Compared with their work, we focus on how to incorporate the corresponding supporting profile information to alleviate ambiguity in a single-turn setting while they adopt the multi-turn interaction manner. Another strand of work Zhang et al. (2018); Zheng et al. (2019); Song et al. (2020) consider incorporating profile information to ease ambiguity and generate consistent dialogue responses. Unlike their work, we focus on the SLU domain while they mainly consider the end-to-end dialogue systems. To the best of our knowledge, this is the first work to consider additional information to alleviate the ambiguity of utterances in the SLU system.

7 Conclusion

In this paper, we investigate the Profile-based SLU, which requires a model to rely not only on the surface utterance but also on the supporting information. We further introduce a large-scale annotated dataset to facilitate further research. In addition, we explore a multi-level knowledge adapter to effectively inject the supporting information. To the best of our knowledge, we are the first to consider Profile-based SLU.

Fusion Methods	Slot (F1)	Intent (Acc)	Overall (Acc)
AGIF (Qin et al. 2020)	37.16	27.12	21.28
AGIF w/Profile	84.18	84.18	77.97
GL-GIN (Qin et al. 2021c)	34.29	25.80	20.90
GL-GIN w/Profile	83.74	86.44	78.34

Table 6: The Performance of Multi-Intent Baselines on the PROSLU dataset.

Fusion Methods	Slot (F1)	Intent (Acc)	Overall (Acc)
Concat	80.08	82.67	76.84
MLP	80.60	83.43	77.78
Hierarchical	**83.24**	**85.69**	**79.28**

Table 7: Ablation Experiments of Different Fusion Methods.

Ethical Considerations

Each sample in our dataset is checked by annotators to ensure the content does not pose potential risks. As indicated in the main text, the annotators are properly paid, and we've taken several steps to both ensure a proper working burden and a high quality dataset.

A Appendix

A.1 Exploration of the Multi-Intent Baselines

We explore the state-of-the-art multi-intent baselines AGIF (Qin et al. 2020) and GL-GIN (Qin et al. 2021c) on the PROSLU dataset. The results are shown in Table 6. We observe that both baselines show poor performance without profile information. With the help of supporting profile information, they improve significantly by a large margin based on our multi-level knowledge adapter.

A.2 Ablation Experiments of Different Fusion Methods

In addition to adopting the hierarchical attention fusion mechanism in Section 4.3, we also try to utilize two traditional fusion layers to aggregate information from different sources as knowledge adapter:

- Concatenation (Concat) is a simple and effective method (Wu et al. 2018) that directly concatenate representation from different sources for each sample and
- Multilayer Perceptron (MLP) can automatically capture the integrated representation (Nguyen and Okatani 2018) which applies an MLP layer on the concatenated output to further abstract the expressive aggregated representations and better extract the multi-source information.

As shown in Table 7, MLP fusion achieves better results than Concat fusion, but underperforms our hierarchical fusion method. This demonstrates that our hierarchical fusion can get word-level dynamic representations of multi-source information and inject them through the multi-level adapter to achieve fine-grained knowledge transfer.

Intent Group
PlayMusic, PlayVideo, PlayAudioBook
SearchMusic, SearchVideo, SearchAudioBook
SearchLocation, SearchLocationOntheway
SearchMetroRoute, SearchBusRoute, SearchDriveRoute
SearchTrainTicket, SearchFlightTicket, SearchCoachTicket

Table 8: Intent groups in the PROSLU dataset.

A.3 General SLU model

Intent Detection Decoder To perform intent detection, a sentence self-attention mechanism (Zhong, Xiong, and Socher 2018) is applied for obtaining sentence representation $\mathbf{g}$, using:

$$\alpha_i = \frac{\exp\left(\mathbf{w}_{\mathrm{g}}^{\top}\mathbf{e}_i\right)}{\sum_j \exp\left(\mathbf{w}_{\mathrm{g}}^{\top}\mathbf{e}_j\right)}, \tag{11}$$

$$\mathbf{g} = \sum\nolimits_i \alpha_i \mathbf{e}_i, \tag{12}$$

where $\mathbf{w}_g \in \mathbb{R}^d$ are trainable model parameters.

Then, $\mathbf{g}$ is used as input for intent detection:

$$\mathbf{y}^{\mathrm{I}} = \mathrm{softmax}\left(\mathbf{W}_{\mathrm{I}}\,\mathbf{g}\right), \tag{13}$$

$$o^{\mathrm{I}} = \arg\max(\mathbf{y}^{\mathrm{I}}), \tag{14}$$

where o^{I} is the predicted intent label; $\mathbf{W}_{\mathrm{I}}$ are trainable parameters.

Slot Filling Decoder We use a unidirectional LSTM as the slot filling decoder. At each decoding step t, we adopt the intent-guided mechanism (Qin et al. 2019) and the decoder hidden state $\mathbf{h}_t^{\mathrm{S}}$ can be formalized as:

$$\mathbf{s}_t = \mathbf{e}_t \oplus \phi^{\text{intent}}\left(o^{\mathrm{I}}\right) \oplus \phi^{\text{slot}}\left(o_{t-1}^{\mathrm{S}}\right), \tag{15}$$

$$\mathbf{h}_t^{\mathrm{S}} = \mathrm{LSTM}\left(\mathbf{s}_t, \mathbf{h}_{t-1}^{\mathrm{S}}\right), \tag{16}$$

where $\mathbf{h}_{t-1}^{\mathrm{S}}$ is the previous decoder state; $\mathbf{e}_t$ is the aligned encoder hidden state and $\mathbf{s}_t$ is the concatenated input for the slot filling decoder; $\phi^{\text{intent}}(\cdot)$ and $\phi^{\text{slot}}(\cdot)$ represent the embedding matrix of intents and slots, respectively. Finally, $\mathbf{h}_t^{\mathrm{S}}$ is used for slot filling:

$$\mathbf{y}_t^{\mathrm{S}} = \mathrm{softmax}\left(\mathbf{W}_{\mathrm{S}}\mathbf{h}_t^{\mathrm{S}}\right), \tag{17}$$

$$o_t^{\mathrm{S}} = \arg\max(\mathbf{y}_t^{\mathrm{S}}), \tag{18}$$

where o_t^{S} is the slot label of the t-th word in the utterance and $\mathbf{W}_{\mathrm{S}}$ are trainable parameters.

Joint Training The intent detection objection is formulated as:

$$\mathcal{L}_{\mathrm{I}} \triangleq -\sum_{i=1}^{n_{\mathrm{I}}} \hat{y}^{i,\mathrm{I}} \log\left(y^{i,\mathrm{I}}\right). \tag{19}$$

Similarly, the slot filling task objection is defined as:

$$\mathcal{L}_{\mathrm{S}} \triangleq -\sum_{t=1}^{T}\sum_{i=1}^{n_{\mathrm{S}}} \hat{y}_t^{i,\mathrm{S}} \log\left(y_t^{i,\mathrm{S}}\right), \tag{20}$$

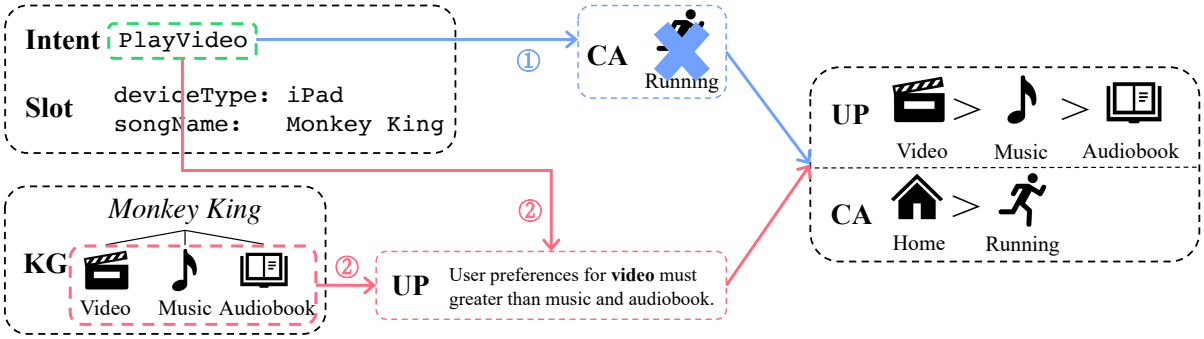

Figure 4: An illustration of generating UP and CA Information.

Input	
Utterance	*Play Monkey King*
KG	`subject`: *Monkey King*, `type`: CreativeWork, `singer`: Mayday, `Year`: 2004, `Description`: ..., `subject`: *Monkey King*, `type`: Movie, `Producer`: Pengcheng Wang, `Writer`: Pengcheng Wang, `Description`: ..., `subject`: *Monkey King*, `type`: CreativeWork, `Category`: Fantasy Novel, `Description`: ...,
UP	`Preference` for [music, video & audiobook]: [0.2, 0.7, 0.1], `Has Car`: True, ...
CA	`Movement State`: stationary, `Posture`: lying down, `Geographic Location`: home, ...
Output	
Intent	`PlayVideo`
Slot	`O B-PlayVideo.videoName I-PlayVideo.videoName`

Table 9: An example in the PROSLU dataset.

where $\hat{y}^{i,\mathrm{I}}$ and $\hat{y}_t^{i,\mathrm{S}}$ are golden intent labels and golden slot labels separately, n_{I} and n_{S} is the number of intent labels and slot labels respectively.

The final joint objective to optimize intent detection and slot filling together is formulated as:

$$\mathcal{L} = \mathcal{L}_{\mathrm{I}} + \mathcal{L}_{\mathrm{S}}. \tag{21}$$

In addition, the shared encoding representations learned by the shared self-attentive encoder can consider two tasks jointly and further ease the error propagation compared with pipeline models (Zhang and Wang 2016) through the final joint loss function.

A.4 Example

Ambiguous Intent Groups The ambiguous intent groups we designed are shown in Table 8. For example, {`PlayMusic`, `PlayVideo`, `PlayAudioBook`} is an ambiguous intent group where each intent is ambiguous with each other ("Play Monkey King" can be any intent in this ambiguous intent group). Slot labels are collected directly from the slot label sets corresponding to the intents.

A Detailed Example in PROSLU We show a detailed example in our PROSLU dataset in Table 9. For the utterance "Play Monkey King", `PlayMusic`, `PlayVideo`, `PlayAudioBook` are possible intents in our intent set. Given the three entities in the KG information, the above intents are all reasonable. Considering the UP information, we can find that the user likes watching videos more than listening to music and audiobook. Finally, the CA information shows that the user is lying at home, which is a reasonable state to watch videos. Therefore, the real intent of the user can be predicted to `PlayVideo`.

An Illustration of Generating UP and CA Information As shown in Figure 4, for the `PlayVideo` intent, the movement state in CA information cannot be running, and if the entity types of entities in the KG information are music, video and audiobook, the user preferences for video is more likely to be greater than music and audiobook.

Acknowledgements

This work was supported by the National Key R&D Program of China via grant 2020AAA0106501 and the National Natural Science Foundation of China (NSFC) via grant 61976072 and 61772153. This work was also supported by the Zhejiang Lab's International Talent Fund for Young Professionals.

References

Bhargava, A.; Celikyilmaz, A.; Hakkani-Tür, D.; and Sarikaya, R. 2013. Easy contextual intent prediction and slot detection. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 8337–8341. IEEE.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating Embeddings for Modeling Multi-relational Data. In Burges, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Chen, W.; Wang, H.; Chen, J.; Zhang, Y.; Wang, H.; Li, S.; Zhou, X.; and Wang, W. Y. 2020. TabFact: A Large-scale Dataset for Table-based Fact Verification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Chen, Y.-N.; Hakkani-Tür, D.; Tür, G.; Gao, J.; and Deng, L. 2016. End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding. In *Interspeech*, 3245–3249.

Chen, Y.-N.; Sun, M.; Rudnicky, A. I.; and Gershman, A. 2015. Leveraging behavioral patterns of mobile applications for personalized spoken language understanding. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, 83–86.

Clark, K.; Luong, M.; Le, Q. V.; and Manning, C. D. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

E, H.; Niu, P.; Chen, Z.; and Song, M. 2019. A Novel Bi-directional Interrelated Model for Joint Intent Detection and Slot Filling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5467–5471. Florence, Italy: Association for Computational Linguistics.

Goo, C.-W.; Gao, G.; Hsu, Y.-K.; Huo, C.-L.; Chen, T.-C.; Hsu, K.-W.; and Chen, Y.-N. 2018. Slot-Gated Modeling for Joint Slot Filling and Intent Prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 753–757. New Orleans, Louisiana: Association for Computational Linguistics.

Hakkani-Tür, D.; Tür, G.; Celikyilmaz, A.; Chen, Y.-N.; Gao, J.; Deng, L.; and Wang, Y.-Y. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Interspeech*, 715–719.

Hemphill, C. T.; Godfrey, J. J.; and Doddington, G. R. 1990. The ATIS Spoken Language Systems Pilot Corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, C.; Li, L.; and Qi, J. 2018. A Self-Attentive Model with Gate Mechanism for Spoken Language Understanding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3824–3833. Brussels, Belgium: Association for Computational Linguistics.

Libovický, J.; and Helcl, J. 2017. Attention Strategies for Multi-Source Sequence-to-Sequence Learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 196–202. Vancouver, Canada: Association for Computational Linguistics.

Liu, B.; and Lane, I. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

Liu, L.; Ren, X.; Shang, J.; Gu, X.; Peng, J.; and Han, J. 2018. Efficient Contextualized Representation: Language Model Pruning for Sequence Labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1215–1225. Brussels, Belgium: Association for Computational Linguistics.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1412–1421. Lisbon, Portugal: Association for Computational Linguistics.

Nguyen, D.-K.; and Okatani, T. 2018. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6087–6096.

Qin, L.; Che, W.; Li, Y.; Wen, H.; and Liu, T. 2019. A Stack-Propagation Framework with Token-Level Intent Detection for Spoken Language Understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2078–2087. Hong Kong, China: Association for Computational Linguistics.

Qin, L.; Che, W.; Ni, M.; Li, Y.; and Liu, T. 2021a. Knowing where to leverage: Context-aware graph convolutional network with an adaptive fusion layer for contextual spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29: 1280–1289.

Qin, L.; Liu, T.; Che, W.; Kang, B.; Zhao, S.; and Liu, T. 2021b. A co-interactive transformer for joint slot filling and intent detection. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8193–8197. IEEE.

Qin, L.; Wei, F.; Xie, T.; Xu, X.; Che, W.; and Liu, T. 2021c. GL-GIN: Fast and Accurate Non-Autoregressive Model for Joint Multiple Intent Detection and Slot Filling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 178–188. Online: Association for Computational Linguistics.

Qin, L.; Xie, T.; Che, W.; and Liu, T. 2021d. A Survey on Spoken Language Understanding: Recent Advances and New Frontiers. In Zhou, Z.-H., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 4577–4584. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Qin, L.; Xu, X.; Che, W.; and Liu, T. 2020. AGIF: An Adaptive Graph-Interactive Framework for Joint Multiple Intent

Detection and Slot Filling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 1807–1816. Online: Association for Computational Linguistics.

Song, H.; Wang, Y.; Zhang, W.-N.; Zhao, Z.; Liu, T.; and Liu, X. 2020. Profile Consistency Identification for Open-domain Dialogue Agents. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6651–6662. Online: Association for Computational Linguistics.

Srinivasan, T.; Sanabria, R.; Metze, F.; and Elliott, D. 2020. Multimodal Speech Recognition with Unstructured Audio Masking. *ArXiv*, abs/2010.08642.

Su, S.-Y.; Yuan, P.-C.; and Chen, Y.-N. 2018. How Time Matters: Learning Time-Decay Attention for Contextual Spoken Language Understanding in Dialogues. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2133–2142. New Orleans, Louisiana: Association for Computational Linguistics.

Teng, D.; Qin, L.; Che, W.; Zhao, S.; and Liu, T. 2021. Injecting Word Information with Multi-Level Word Adapter for Chinese Spoken Language Understanding. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8188–8192.

Tsai, Y.-H. H.; Bai, S.; Liang, P. P.; Kolter, J. Z.; Morency, L.-P.; and Salakhutdinov, R. 2019. Multimodal Transformer for Unaligned Multimodal Language Sequences. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6558–6569. Florence, Italy: Association for Computational Linguistics.

Tur, G.; and De Mori, R. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Wang, Y.; Shen, Y.; and Jin, H. 2018. A Bi-Model Based RNN Semantic Frame Parsing Model for Intent Detection and Slot Filling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 309–314. New Orleans, Louisiana: Association for Computational Linguistics.

Wu, Z.; Dai, X.-Y.; Yin, C.; Huang, S.; and Chen, J. 2018. Improving review representations with user attention and product attention for sentiment classification. In *Thirty-second AAAI conference on artificial intelligence*.

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

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, 5753–5763.

Young, S.; Gašić, M.; Thomson, B.; and Williams, J. D. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5): 1160–1179.

Zadeh, A.; Chen, M.; Poria, S.; Cambria, E.; and Morency, L.-P. 2017. Tensor Fusion Network for Multimodal Sentiment Analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1103–1114. Copenhagen, Denmark: Association for Computational Linguistics.

Zhang, S.; Dinan, E.; Urbanek, J.; Szlam, A.; Kiela, D.; and Weston, J. 2018. Personalizing Dialogue Agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2204–2213. Melbourne, Australia: Association for Computational Linguistics.

Zhang, X.; and Wang, H. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2993–2999.

Zheng, Y.; Chen, G.; Huang, M.; Liu, S.; and Zhu, X. 2019. Personalized dialogue generation with diversified traits. *arXiv preprint arXiv:1901.09672*.

Zhong, V.; Xiong, C.; and Socher, R. 2018. Global-Locally Self-Attentive Encoder for Dialogue State Tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1458–1467. Melbourne, Australia: Association for Computational Linguistics.

Zhu, S.; Cao, R.; and Yu, K. 2020. Dual learning for semi-supervised natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28: 1936–1947.