Decoupling Breaks Data Barriers: A Decoupled Pre-training Framework for Multi-Intent Spoken Language Understanding

Libo Qin[1] , **Qiguang Chen**[2] , **Jingxuan Zhou**[1] , **Qinzheng Li**[1] , **Chunlin Lu**[1] and **Wanxiang Che**[2]
[1] School of Computer Science and Engineering, Central South University, China
[2] Research Center for Social Computing and Information Retrieval
[2] Harbin Institute of Technology, China
lbqin@csu.edu.cn, {qgchen,car}@ir.hit.edu.cn

Abstract

Multi-intent Spoken Language Understanding (Multi-intent SLU) can extract multiple intents in a single utterance, gaining increasing attention. Nevertheless, current multi-intent SLU approaches still heavily rely on large amounts of annotated multi-intent SLU data, which makes it hard to be satisfied in real-world scenarios without sufficient data. Motivated by this, we introduce a novel decoupled pre-training framework (DPF) to address the data-scarcity problem, achieving to leverage of abundant multi-intent-free SLU data to enhance multi-intent SLU. Specifically, DPF first decouples the multi-intent SLU task into two abilities: (1) *task-agnostic ability* to locate the task-agnostic slot entity span and (2) *task-specific ability* to predict the task-specific slot and intent labels simultaneously. The key insight of DPF is that such decomposition allows us to design a two-stage decoupled pre-training procedure to enhance both *task-agnostic ability* and *task-specific ability* with abundant multi-intent-free SLU data (i.e., NER and single-intent SLU data), respectively. Experimental results on two standard benchmarks (e.g., MixATIS and MixSNIPS) demonstrate the effectiveness of DPF by achieving superior performance. In addition, extensive analyses reveal that utilizing the multi-intent-free data can effectively enhance multi-intent SLU.

1 Introduction

Spoken Language Understanding (SLU) is an essential component of dialog systems, which can be used to extract the semantic parsing results of user query (e.g., intents and slots) [Tur and De Mori, 2011; Young *et al.*, 2013; Qin *et al.*, 2019]. Specifically, SLU consists of two typical sub-tasks: slot filling and intent detection. Take the query "*add Despacito to POP playlist and play it*" in the Figure 1 as an example, the former task is regarded as a sequence labeling task to output a sequence of slots (i.e., "O, B-music-name, O, B-playlist, O, O, O, O") and the latter task is considered as a classification task to predict intents (i.e., "AddToPlaylist, PlayMusic").

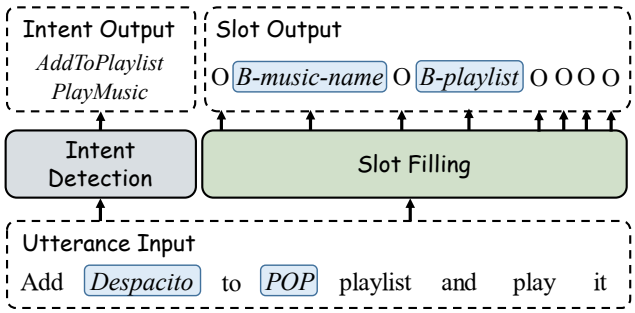
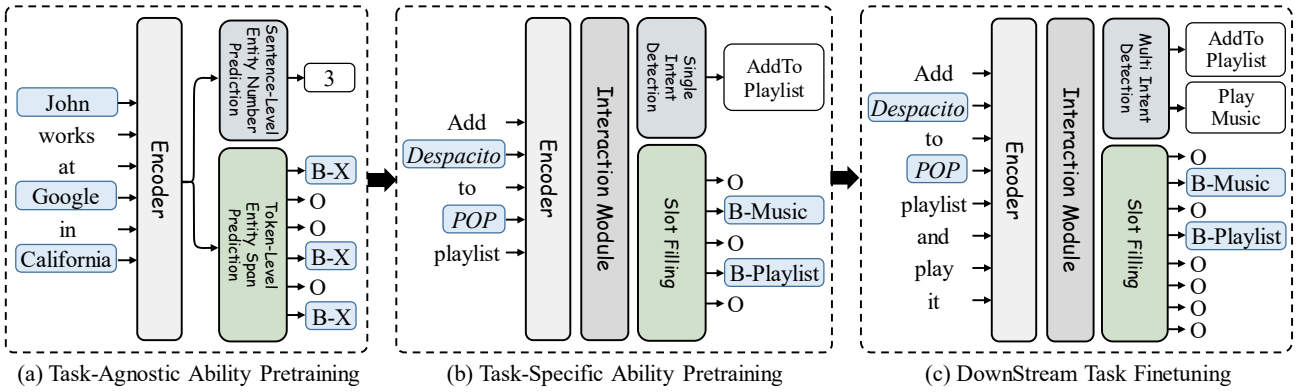


Figure 1: An example of multi-intent SLU, which consists of a sequence of slots and multiple intents.

Recently, Gangadharaiah & Narayanaswamy*et al.* [2019] discover that over 50% of samples in the internal Amazon dataset exhibit multiple intents. Consequently, dominant SLU systems shift their eyes from single-intent settings to multi-intent scenarios [Qin *et al.*, 2020; Wu *et al.*, 2022; Song *et al.*, 2022b]. To this end, Qin *et al.* [2020] introduce AGIF, a method that adaptively integrates information from multi-intent predictions into the slot filling process. Song *et al.* [2022b] incorporate statistical information regarding the co-occurrence frequencies of intents and slots to improve multi-intent SLU. Xing & Tsang *et al.* [2022] propose a heterogeneous semantics-label graphs framework to model the mutual guidance between the two tasks. Pham *et al.* [2023] introduce MISCA, a joint model with intent-slot and label attention mechanisms to capture intent-slot correlations, achieving promising performance.

However, despite its success, current dominant multi-intent SLU models still heavily rely on a large amount of annotated data for training, which poses a significant cost burden for the real-world scenario without sufficient data. In addition, annotating multi-intent data is extremely time-consuming and labor-intensive, because annotators are not only required to have professional knowledge in task-oriented dialogue systems but also to understand the complex multi-intent relationships within a user query. Therefore, in this work, we aim to



(a) Task-Agnostic Ability Pretraining (b) Task-Specific Ability Pretraining (c) DownStream Task Finetuning

Figure 2: The overall workflow for DPF, which consists of (a) task-agnostic ability pretraining, (b) task-specific ability pretraining, and (c) downstream task finetuning.

investigate a research question "***Can we leverage the readily accessible multi-intent-free data, such as NER and single-intent SLU data, to enhance multi-intent SLU?***".

Motivated by this, in this paper, we introduce a novel decoupled pre-training framework (DPF) for multi-intent SLU, aiming to utilize abundant multi-intent-free data (i.e., NER and single-intent SLU data). Specifically, DPF decouples multi-intent SLU into two abilities: (1) the *task-agnostic ability* to identify the slot entity span and (2) the *task-specific ability* to predict domain-specific slot and intent labels simultaneously. Such decomposition allows us to design a two-stage pre-training framework to enhance the two abilities, respectively, which is shown in Figure 2. Concretely, to enhance the *task-agnostic ability*, DPF first proposes a task-agnostic ability pre-training stage. This stage contains a token-level entity span prediction pre-training task and sentence-level entity number prediction pre-training task, facilitating representation learning for token-level slot filling and sentence-level intent detection. Since the task-agnostic ability pre-training stage does not require any task-specific annotation data, this process can be achieved with abundant NER data (see Figure 2 (a)). To improve the *task-specific ability*, DPF further proposes a task-specific ability pre-training stage, which mainly focuses on learning the mutual guidance between intent detection and slot filling by leveraging single-intent SLU data (see Figure 2 (b)). Finally, after completing the two-stage pre-training procedure, we achieve the multi-intent SLU by conducting the downstream task finetuning (see Figure 2(c)).

With the help of decoupled pre-training, it brings us at least two advantages: (1) By disentangling the multi-intent SLU into task-agnostic and task-specific ability learning, our model is able to sequentially solve multi-intent SLU step by step; (2) In contrast to previous approaches that heavily rely on multi-intent SLU training data, DPF can leverage a large amount of NER and single-intent SLU data, which are much easier to obtain.

Contributions of this work are summarized as:

(1) To the best of our knowledge, this study is the first to investigate a decoupled pre-training framework (DPF) for multi-intent SLU;

(2) DPF offers the advantage of utilizing a vast amount of multi-intent-free data to enhance multi-intent SLU, thus effectively alleviating the data scarcity problem;

(3) Experimental results show that DPF achieves the superior performance and extensive analyses reveal its superiority in low-resources scenarios.

To facilitate the research, all codes used in this work are publicly available at https://github.com/LightChen233/DPF.

2 Methodology

This section provides the overall workflow of DPF, which consists of task-agnostic ability pretraining stage (TAAP) (§2.1), a task-specific ability pretraining stage (TSAP) (§2.2) and downstream task finetuning (DSTF) (§2.3). Specifically, Algorithm 1 shows the overall process of DPF. Lines 1-4 denote the two-stage pretraining procedures including task-agnostic and task-specific ability pretraining. Lines 5-7 denote the fine-tuning stage. We will describe each stage in the following sub-sections.

2.1 Stage 1: Task-Agnostic Ability Pretraining

Task-Agnostic Ability Pretraining (TAAP) is used to enhance the general task-agnostic ability of multi-intent SLU that does not require any task-oriented dialogue domain knowledge. Therefore, we can leverage abundant NER data for the task-agnostic ability pre-training. Since multi-intent SLU contains token-level slot filling and sentence-level intent detection sub-tasks, we introduce *token-level entity span prediction pretraining task* and *sentence-level entity number prediction pretraining task* to capture the token-level and sentence-level general knowledge, respectively.

Token-Level Entity Span Prediction

Token-level entity span prediction pretraining task can be used to capture token-level task-agnostic ability. Specifically, it is considered as a binary task of token-level entity span prediction to judge whether a token belongs to an entity span or not. As shown in Figure 2 (a), take the input sentence *"John works at Google in California."* as an example, the entities in this sentence are labeled as follows: *"John [B-Person] works*

Algorithm 1 Overall Workflow of DPF

Input: Dataset $\mathcal{D} = \{(task, x, y)_i\}_{i=1}^{|\mathcal{D}|}$; The number of trained epochs in each tasks $e_{\text{max}}^{\text{task}}$; Initial model parameters Θ.
Output: Trained Model Θ.

```
// decoupled pretraining for TAAP and TSAP
for task in {TAAP, TSAP} do
    for e in {1, ..., e_max^task} do
        // batch optimization
        for B in {(task, x, y)_j}_{j=1}^{|B|} ∈ D do
            Optimizing Θ using L_task;
// downstream task finetuning
for e in 1, ..., e_max^DSTF do
    for B in {(DSTF, x, y)_j}_{j=1}^{|B|} ∈ D do
        Optimizing Θ using L_DSTF;
```

at Google [B-Company] in California [B-Location]", we use the sketch to replace the original entities annotation as "*John [B-X] works at Google [B-X] in California [B-X]*" to perform an entity binary prediction task.

Formally, given input sentence $X = \{\texttt{[CLS]}, x_1, x_2, \ldots, x_n\}$, we first encode X to obtain hidden vector $\mathbf{H} = \{h_{\texttt{[CLS]}}, h_1, h_2, \ldots, h_n\}$, where n is the sequence length:

$$\mathbf{H} = \texttt{Encoder}(X), \tag{1}$$

where we use Deberta$_{v3}$ [He *et al.*, 2023] as the `Encoder` in this work.

Furthermore, given the $\mathbf{H}$, *token-level entity span prediction pretraining task* is used to output the entity sequence $E = \{e^1, e^2, \ldots, e^n\}$, where $e^* \in \{$*B-X*, *I-X*, *O*$\}$, which is denoted as:

$$e^j = \texttt{softmax}(\mathbf{W}^E \mathbf{H}^j + \mathbf{b}), \tag{2}$$

where the $\mathbf{W}^E$ and $\mathbf{b}$ are learnable parameters.

The objective of the token-level entity span prediction task is formulated as:

$$\mathcal{L}_{TL} = -\frac{1}{3n} \sum_{i=1}^{3} \sum_{j=1}^{n} \hat{e}^j \log e^j + (1 - \hat{e}^j) \log(1 - e^j), \tag{3}$$

where $\hat{e}^j$ refers to the gold entity label at j token.

Sentence-Level Entity Number Prediction

To enhance the sentence-level task-agnostic ability, we introduce a *sentence-level entity number prediction pre-training task* to predict the number of entities in an utterance, which achieves to improve the model's understanding ability of the whole sentence. The underlying intuition is that by accurately predicting the number of entities in a sentence, the model can successfully capture the entire sentence representation, which is beneficial for sentence-level intent detection.

Similarly, as shown in Figure 2 (b), given the input utterance *"John works at Google in California."* the number of entities is determined to be "3" (i.e., "*John [B-X]*", "*Google [B-X]*", and "*California [B-X]*"). Formally, the sentence representation $h_{\texttt{[CLS]}}$ is used to predict the number m of entities in an utterance, which is denoted as:

$$m = \texttt{softmax}(\mathbf{W} h_{\texttt{[CLS]}} + \mathbf{b}). \tag{4}$$

The training objective can be defined as follows:

$$\mathcal{L}_{SL} = -\frac{1}{|m|} \sum_{i=0}^{|m|} \hat{m} \log m + (1 - \hat{m}) \log(1 - m), \tag{5}$$

where $|m|$ is the maximum number of entities and $\hat{m}$ denotes the golden entity number.

Joint Pretraining

We adopt a joint training method for *token-level entity span prediction pretraining task* and *sentence-level entity number prediction pre-training task* simultaneously, denoting as:

$$\mathcal{L}_{\text{TAAP}} = \alpha_1 \mathcal{L}_{TL} + \alpha_2 \mathcal{L}_{SL}, \tag{6}$$

where α_1 and α_2 are hyper-parameters.

Pre-training Data Collection

The task-agnostic ability pretraining stage does not require the task-specific annotation about slot and intent, which allows us to leverage large amounts of NER data. Therefore, we collect various NER datasets, including BBN [Weischedel and Brunstein, 2005], CoNLL [Sang and De Meulder, 2003], FIGER [Ling and Weld, 2012], GUM [Zeldes, 2017], SLURP [Bastianelli *et al.*, 2020], MITMovieCorpus [Liu *et al.*, 2013], MITRestaurantCorpus [Liu *et al.*, 2013], MultiCoNER [Fetahu *et al.*, 2021], MultiNERD [Tedeschi and Navigli, 2022], OntoNotes 5.0 [Pradhan *et al.*, 2013], Polyglot-NER [Al-Rfou *et al.*, 2015], Ritter [Ritter *et al.*, 2011], WikiANN [Pan *et al.*, 2017], WikiNEuRal [Tedeschi *et al.*, 2021], WNUT17 [Derczynski *et al.*, 2017], XGLUE [Liang *et al.*, 2020].

Overall, we collect 2.4 million training data for the task-agnostic ability pre-training procedure. The detailed data statistics are shown in Table 1.

Dataset	Train	Dev	Test
BBN [Weischedel and Brunstein, 2005]	33K	-	6K
CoNLL [Sang and De Meulder, 2003]	15K	3K	4K
FIGER [Ling and Weld, 2012]	1.2M	10k	0.3K
GUM [Zeldes, 2017]	2K	-	1K
SLURP [Bastianelli *et al.*, 2020]	12K	2K	3K
MITMovieCorpus [Liu *et al.*, 2013]	8K	-	2K
MITRestaurantCorpus [Liu *et al.*, 2013]	8K	-	2K
MultiCoNER† [Fetahu *et al.*, 2021]	15K	0.8K	218K
MultiNERD† [Tedeschi and Navigli, 2022]	164K	-	-
OntoNotes 5.0† [Pradhan *et al.*, 2013]	60K	9K	8K
Polyglot-NER† [Al-Rfou *et al.*, 2015]	424K	-	-
Ritter [Ritter *et al.*, 2011]	2K	-	-
WikiANN† [Pan *et al.*, 2017]	20K	10K	10K
WikiNEuRal† [Tedeschi *et al.*, 2021]	93K	12K	12K
WNUT17 [Derczynski *et al.*, 2017]	3K	1K	1K
XGLUE [Liang *et al.*, 2020]	14K	3K	3K

Table 1: Statistics of all datasets used in TAAP stage, where † denotes that we only use the English split in those multi-lingual datasets.

Dataset	Train	Dev	Test
ATIS [Hemphill *et al.*, 1990]	4K	0.5K	-
Facebook† [Schuster *et al.*, 2019]	23K	4K	7K
HWU64 [Liu *et al.*, 2021]	26K	-	-
Leyzer† [Sowański and Janicki, 2020]	2K	0.4K	1K
MMNLU2022† [FitzGerald *et al.*, 2023]	17K	-	-
SNIPS [Coucke *et al.*, 2018]	13K	0.7K	-
xSID† [van der Goot *et al.*, 2021]	37K	0.3K	0.5K

Table 2: Statistics of all datasets used in TSAP stage, where † denotes that we only use the English split in those multi-lingual datasets.

2.2 Stage 2: Task-Specific Ability Pretraining

The core challenge of multi-intent SLU lies in effectively modeling interaction across intents and slots. Unfortunately, there is no sufficient annotated data to capture the interaction relationship in multi-intent SLU. Therefore, we introduce the task-specific ability pretraining (TSAP) stage to leverage relatively abundant single-intent SLU data to enhance the task interaction ability.

Specifically, we adopt an interaction module from the strong single-intent SLU model (DCA-Net) [Qin *et al.*, 2021a] as our pretraining interaction module, which is used to learn task-specific interaction knowledge across slot filling and intent detection. Formally, given the hidden states $\mathbf{H}$, TSAP considers the interaction between slot filling and intent detection to get the interaction representations $\mathcal{I} = \{\mathcal{I}_{[\mathcal{CLS}]}, \mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_n\}$:

$$\mathcal{I} = \texttt{Interaction-Module}(\mathbf{H}). \tag{7}$$

Furthermore, the interaction representations $\mathcal{I}$ are used to perform slot filling and single intent detection in TSAP, which are denoted as:

$$S = \texttt{softmax}(\mathbf{W}\mathcal{I} + \mathbf{b}), \tag{8}$$

$$I = \texttt{softmax}(\mathbf{W}\mathcal{I}_{[\text{CLS}]} + \mathbf{b}). \tag{9}$$

The training loss of the model is similar to Equation 6, which is the weighted sum of slot and intent loss. Finally, the pretrained interaction module can be strengthened by the task-specific ability pretraining stage and is directly used in the downstream task fine-tuning process.

Pre-training Data Collection

For task-specific ability pre-training, we collect several single-intent SLU benchmarks that are easier to obtain compared to multi-intent SLU data, including ATIS [Hemphill *et al.*, 1990], Facebook [Schuster *et al.*, 2019], HWU64 [Liu *et al.*, 2021], Leyzer [Sowański and Janicki, 2020], MMNLU2022 [FitzGerald *et al.*, 2023], SNIPS [Coucke *et al.*, 2018] and xSID [van der Goot *et al.*, 2021].

Finally, we collect 136K samples for TSAP stage. The detailed statistics are shown in Table 2.

2.3 Down-Stream Task Finetuning

After completing the task-agnostic and task-aware ability pretraining, we directly fine-tune the pretrained model for the multi-intent SLU task. During the downstream task finetuning (DSTF) stage, multiple intent detection and slot filling can be simultaneously optimized by multi-task learning.

3 Experiments

3.1 Implementation Settings

We evaluate DPF using two standard benchmarks: MixATIS and MixSNIPS [Qin *et al.*, 2020]. The batch sizes for all experiments are selected from $\{8, 16, 32\}$, and the learning rates are set within the range $[1 \times 10^{-6}, 7 \times 10^{-5}]$. We employ AdamW [Loshchilov and Hutter, 2019] for training our model with a weight decay parameter set to 1×10^{-8}. All experiments are conducted on V100 16G and V100 32G. All models are selected from the development set and evaluated on the test set.

3.2 Baselines

We compare our approach with the following non-pretrained models and pretrained models. Non-pretrained baselines include: AGIF [Qin *et al.*, 2020] implements an adaptive graph network to facilitate detailed multi-intent information interactions; GL-GIN [Qin *et al.*, 2021b] employs a non-autoregressive architecture to accelerate decoding in simultaneous multiple intent detection and slot filling tasks; SDJN [Chen *et al.*, 2022a] presents three interconnected decoders and a self-distillation technique to establish coherence between multiple intents and slots; GIS-Co [Song *et al.*, 2022b] utilizes statistical co-occurrence frequencies of intents and slots as prior knowledge, significantly improving the performance; Co-Guiding [Xing and Tsang, 2022] introduces a framework based on heterogeneous semantics-label graphs, gradually establishing and modeling mutual guidance between intent detection and slot filling.

The pretrained models containing DCA-Net [Qin *et al.*, 2021a] effectively utilize bidirectional, task-specific interaction for simultaneous slot filling and intent detection; TFMN [Chen *et al.*, 2022b] utilizes an encoder for multi-grain representations, enhancing the utterance understanding; DeBERTa$_{v3}$ [He *et al.*, 2023] incorporates a disentangled attention mechanism in pre-training to improve representation quality; MISCA [Pham *et al.*, 2023] introduces a dual mechanism of intent-slot co-attention and label attention, boosting multi-task interaction; DGIF [Zhu *et al.*, 2023] uses label semantic information as enriched priors and constructs a multi-grain interactive graph to map intent-slot correlations; MTLN-GP [Wan *et al.*, 2023] presents a multi-dimensional type-slot label interaction network coupled with a global pointer network for efficient handling of nested and non-nested slots and slot incoherence, leading to quicker inference. Besides, we replace the encoders of non-pretrained models with DeBERTa$_{v3}$ for fair comparison. Baseline results are token from previous works [Chen *et al.*, 2022a; Chen *et al.*, 2022b; Song *et al.*, 2022b; Xing and Tsang, 2022; Zhu *et al.*, 2023; Wan *et al.*, 2023]. The reproduced results are based on OpenSLU [Qin *et al.*, 2023] framework.

3.3 Main Results

Following the settings of Goo *et al.* [2018] and Qin *et al.* [2020], we evaluate the performance of multi-intent SLU with three metrics: F1 score for slot filling (Slot F1), accuracy score for intent detection (Intent Acc.), and the exact match

Model	MixSNIPS			MixATIS		
	EMA (%)	Slot F1.(%)	Intent Acc.(%)	EMA %)	Slot F1.(%)	Intent Acc.(%)
Non-Pretrained Models						
AGIF [Qin *et al.*, 2020]	74.2	94.2	95.1	40.8	86.7	74.4
GL-GIN [Qin *et al.*, 2021b]	75.4	94.9	95.6	43.5	88.3	76.3
SDJN [Chen *et al.*, 2022a]	75.7	94.4	96.5	44.6	88.2	77.1
GIS-Co [Song *et al.*, 2022b]	75.9	-	-	48.2	-	-
Co-Guiding [Xing and Tsang, 2022]	77.5	95.1	97.7	51.3	89.8	79.1
Pretrained Models						
DeBERTa$_{v3}$‡ [He *et al.*, 2023]	80.4	95.9	96.4	44.7	83.7	76.9
DeBERTa$_{v3}$+ AGIF‡ [Qin *et al.*, 2020]	83.5	95.6	96.7	45.4	87.0	75.2
DeBERTa$_{v3}$+ DCA-Net‡ [Qin *et al.*, 2021a]	83.1	95.6	96.2	47.0	81.8	76.7
DeBERTa$_{v3}$+ GL-GIN‡ [Qin *et al.*, 2021b]	83.8	96.4	96.9	47.5	84.4	79.1
DeBERTa$_{v3}$+ Co-Guiding‡ [Xing and Tsang, 2022]	85.6	97.4	96.9	48.4	85.0	78.6
DeBERTa$_{v3}$+ GIS-Co‡ [Song *et al.*, 2022b]	82.5	96.4	96.5	44.3	84.5	78.7
TFMN [Chen *et al.*, 2022b]	84.7	96.4	97.7	50.2	88.0	79.8
DeBERTa$_{v3}$+ MISCA‡ [Pham *et al.*, 2023]	83.0	96.4	96.7	45.1	84.0	78.0
MTLN-GP [Wan *et al.*, 2023]	84.3	96.7	97.9	49.4	88.4	79.6
DGIF [Zhu *et al.*, 2023]	84.3	95.9	97.8	50.7	88.5	**83.3**
DPF	**93.1**	**98.7**	**98.0**	**55.4**	**90.4**	80.9

Table 3: Main Results. ‡ denotes that we reproduce those models based on DeBERTa$_{v3}$ backbone.

Model	EMA (%)	Slot F1.(%)	Intent Acc.(%)
MixATIS			
Our	**55.4**	**90.4**	**80.9**
w/o TAAP	49.3	89.3	77.9
w/o TSAP	48.8	88.5	78.9
w/o TAAP & TSAP	47.0	81.8	76.7
MixSNIPS			
Our	**93.1**	**98.7**	**98.0**
w/o TAAP	86.3	95.9	97.5
w/o TSAP	83.4	96.3	96.0
w/o TAAP & TSAP	83.1	95.6	96.2

Table 4: Ablation Experiments. TAAP and TSAP denote the *task-agnostic ability pretraining* and *task-specific ability pretraining*, respectively.

accuracy (EMA). The results are presented in Table 3. Our observations are as follows:

(1) ***Pretrained models beat most of non-pretrained approaches.*** As illustrated in Table 3, we observe that after replacing the encoders of the models with DeBERTa$_{v3}$, the performance of most of the models improves substantially, demonstrating that the knowledge learned by the pre-trained models can be used to enhance the multi-intent SLU.

(2) ***DPF remarkably improves multi-intent SLU performance.*** As illustrated in Table 3, DPF significantly outperforms all baselines on two benchmarks, including both pre-trained and non-pre-trained models. Specifically, on MixATIS dataset, DPF outperforms the DGIF model by 4.7% on EMA, while on MixSNIPS dataset, it surpasses Deberta$_{v3}$+Co-guiding by 7.5% on EMA, which verifies the effectiveness of DPF.

3.4 Analysis

In this section, we conduct comprehensive analyses to answer the following questions to better understand our approach: (1) Does TAAP capture task-agnostic knowledge? (2) Does TSAP capture task-specific knowledge? (3) Can TAAP and TSAP help each other? (4) What are the impacts of the decoupled pre-training paradigm? (5) Can DPF generalize well on few-shot setting? (6) What is the performance of ChatGPT? (7) Why DPF works?

Answer1: TAAP can Capture both Sentence-level and Token-level Task-agnostic Knowledge

To verify the effectiveness of task-agnostic ability pretraining (TAAP), we remove the procedure of TAAP and only keep the task-specific pre-training unchanged, which we refer to as w/o TAAP.

As shown in Table 4, we observe that the removal of TAAP leads to a decrease in Slot F1 drops by 1.1% for MixATIS and 2.8% for MixSNIPS. Additionally, Intent Acc. declines by 3.0% on MixATIS and drops by 0.5% on MixSNIPS, which demonstrates the effectiveness of TAAP. We attribute it to the fact that the incorporated sentence-level entity num prediction and token-level entity span detection can inject more task-general knowledge into pre-training procedure, which the previous work can not achieve.

Answer2: TSAP can Boost Task-aware Interaction across Intent Detection and Slot Filling

To demonstrate the effectiveness of task-specific ability pretraining (TSAP), we only keep the task-agnostic ability pretraining and directly use it for fine-tuning on downstream tasks. We refer it to the w/o TSAP.

The results are shown in Table 4 (w/o TSAP). We find that EMA drops significantly: 6.6% on MixATIS and 9.7% on MixSNIPS. We suppose that the TSAP can improve the task-aware interaction ability across the two related tasks, which is crucial for the multi-intent SLU.

Answer3: Combination of TAAP and TSAP Brings Further Improvement

To verify the effectiveness of the combination usage of TAAP and TSAP, we remove the two-stage pretraining and directly fine-tune the model with multi-intent SLU data.

Table 4 shows that when both TAAP and TSAP are removed, the overall performance suffers a further decline.

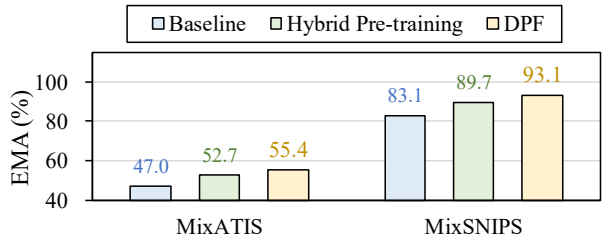
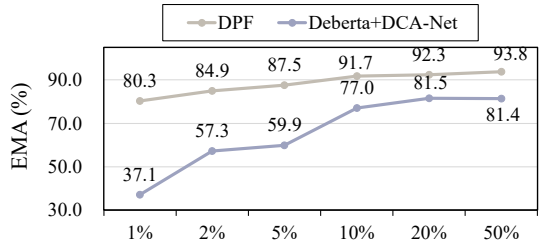
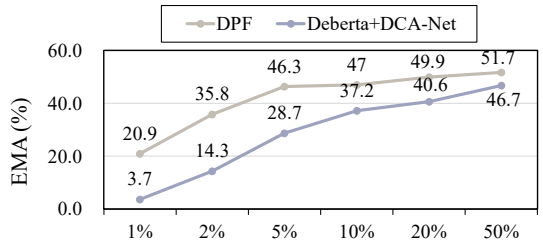


Figure 3: Multi-intent SLU results for data effectiveness analysis on EMA, where "Baseline" denotes "Deberta$_{v3}$+DCA-Net".



(a) EMA on MixSNIPS on Low-Resource Setting.



(a) EMA on MixATIS on Low-Resource Setting.

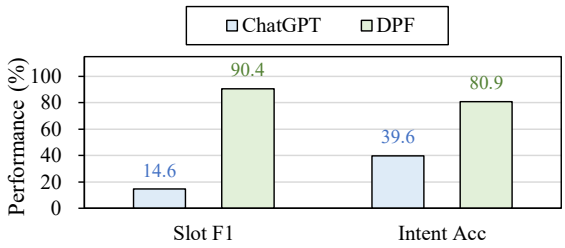
Figure 4: Low-Resource Performance.

Specifically, the model only obtains 47.0% EMA on MixATIS and 83.1% EMA on MixSNIPS, which indicates that the two pretraining stages are complementary to each other and can better facilitate multi-intent SLU.

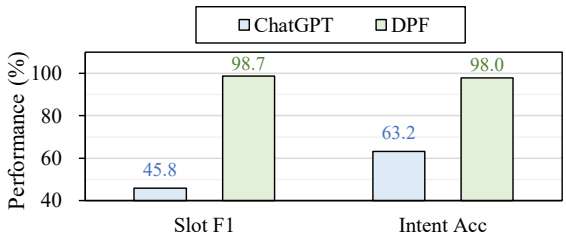
Answer4: Decoupled Pre-training Paradigm rather than Pre-training Data Matters in DPF

It is natural to wonder whether the final performance is primarily attributed to the collected pre-training data or the proposed two-stage decoupled pre-training paradigm. To explore the question, we conduct an experiment by collecting all pre-training data from two stages and only employing one stage pre-training approach (named as Hybrid Pre-training). The loss is the sentence-level and token-level joint loss similar to Equation 6.

The results are shown in Figure 3. We have the following observations: (1) Hybrid Pre-training outperforms the baselines on two benchmarks, which indicates the pre-training procedure can benefit the multi-intent SLU task. (2) DPF beats hybrid Pre-training on the MixATIS and MixSNIPS, with improvements of 2.7% and 3.4% respectively. This confirms that the improvement comes from the introduced two-stage pre-training paradigm rather than the incorporated pre-training data.



(a) The intent and slot performance on MixATIS.



(b) The intent and slot performance on MixSNIPS.

Figure 5: ChatGPT vs. DPF.

Answer5: DPF Works Better on Few-Shot Settings

In order to demonstrate the effectiveness of our model on low-resource settings, we only select 1% to 50% of the original data for fine-tuning.

The results are illustrated in Figure 4. From the results, we observe that DPF beats all baselines on few-shot settings, which indicates that DPF can work in low-resource scenarios. We attribute it to the fact that knowledge captured from the pre-training procedure can be transferred to the low-resource setting, which is consistent with the previous observation [Gururangan *et al.*, 2020].

Answer6: Investigation of ChatGPT

Recently, large language models (e.g., ChatGPT[1]) have dominated the performance in the NLP literature. A natural question arises: Can ChatGPT excellently address the multi-intent SLU task? To answer this question, in this experiment, we utilize the prompt from Pan *et al.* [2023] to investigate ChatGPT for multi-intent SLU.

The comparison results between ChatGPT and DPF are illustrated in Figure 5. We observe that DPF outperforms ChatGPT on MixATIS and MixSNIPS across all metrics, which demonstrates that simply relying on ChatGPT is not sufficient to fully solve the complex multi-intent SLU problem.

Answer7: Qualitative Analysis

To provide a more intuitive understanding of the model, we present a case study that includes the results of three different scenarios: (1) solely using *task-agnostic ability pretraining*, (2) solely using *task-specific ability pretraining*, and (3) simultaneously using both *task-agnostic ability pretraining* and *task-specific ability pretraining* (DPF).

The case study is shown in Figure 6. Take the example as shown in Figure 6 (a) and Figure 6 (b), it fails to yield accurate predictions when either the *task-agnostic ability pretraining* or *task-specific ability pretraining* stage is retained

[1]https://platform.openai.com

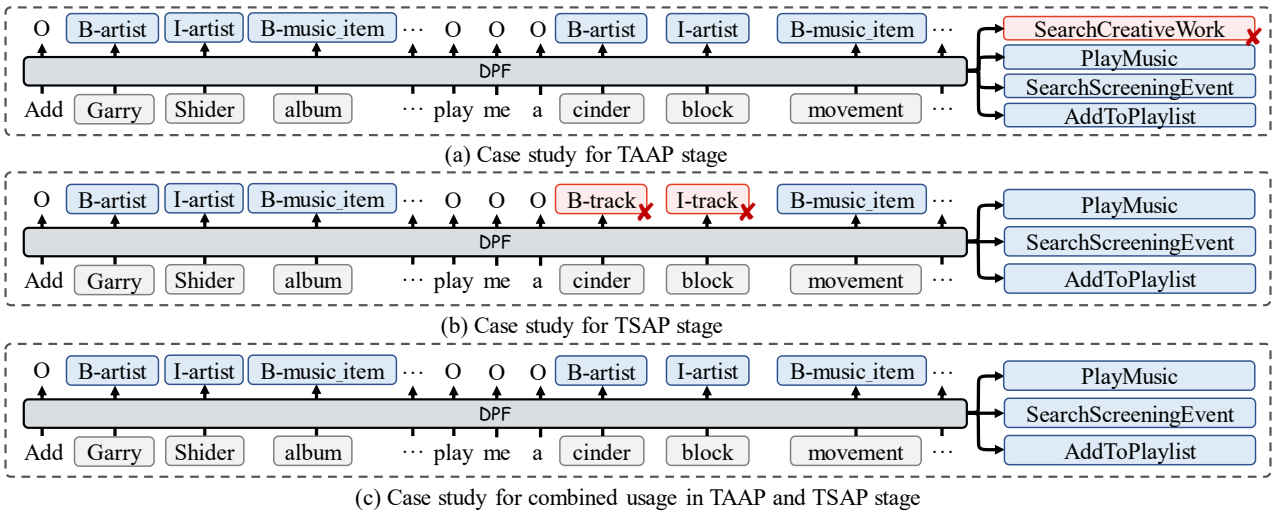


Figure 6: Case study. Texts with blue boxes denote the correct prediction while texts with red boxes stand for the wrong prediction.

alone, and can only be predicted correctly with combined usage, as depicted in Figure 6 (c). We attribute it to the fact that only employing TAAP ignores the interaction across slot and intent, causing the intent prediction wrongly while solely using TSAP stage cannot capture the token-level knowledge for slot filling. In contrast, DPF can capture the interaction between intent and slot, as well as token-level knowledge for slot filling, which brings improvement.

4 Related Work

Multi-intent Spoken Language Understanding (Multi-intent SLU) has gained growing interest due to its ability to discern multiple intents from a given utterance [Qin *et al.*, 2021c]. Motivated by this, researchers shift their focus from single-intent SLU to multi-intent SLU. To this end, Gangadharaiah & Narayanaswamy [2019] first investigate a joint modeling technique for multi-intent SLU. Qin *et al.* [2020] introduce an adaptive graph network that enhances the integration of intent data for detailed slot filling. More importantly, they release two benchmarks to facilitate the multi-intent SLU community. Qin *et al.* [2021b] subsequently introduce a non-autogressive framework to improve the decoding speed. Chen *et al.* [2022b] develop a transformer-based model called TFMN, which adds an auxiliary intent number detection task to improve the model performance. Song *et al.* [2022b] investigate the utilization of statistical co-occurrence frequencies between intents and slots for multi-intent SLU interaction. Similarly, Xing & Tsang*et al.* [2022] propose a novel framework based on heterogeneous semantics-label graphs for multi-intent SLU. Wu *et al.* [2022] and Song *et al.* [2022a] explore the method of prompt-based generative framework. Pham *et al.* [2023] present MISCA, a joint model using intent-slot and label attention mechanisms to capture correlations without any additional graphs. Cheng *et al.* [2023a] propose MRRL framework, which refines output based on references with reinforcement learning. Cheng *et al.* [2023b] introduce TKDF, improving student models via knowledge distillation with evaluator and curriculum learning, which achieves promising performance. Zhu *et al.* [2023] propose DGIF model to leverage label semantics as enriched priors, building a multi-layer interactive graph for intent-slot correlation analysis. Concurrently, Wang *et al.* [2023] present MTLN-GP framework to address both nested and non-nested slot challenges, which can significantly improve the inference speed and performance.

Though the above approaches achieve promising performance, their models still heavily rely on a large amount of annotated multi-intent SLU data. In contrast, our work introduces a decoupled pre-training framework for multi-intent SLU, which allows the model to leverage large amounts of multi-intent-free data, such as NER and single-intent SLU data. To the best of our knowledge, this study represents the first investigation into the utilization of additional multi-intent-free data to enhance multi-intent SLU.

5 Conclusion

In this paper, we introduce a decoupled pretraining framework (DPF), achieving to leverage abundant multi-intent-free data to enhance multi-intent SLU. Specifically, DPF first decouples multi-intent SLU into task-agnostic and task-aware abilities. Furthermore, DPF introduces a two-stage pre-training paradigm to enhance the two abilities, respectively. Experimental results show that DPF achieves superior performance on the MixSNIPS and MixATIS datasets and can also successfully generalize to low-resource settings.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) via grant 62306342, 62236004 and 61976072. Libo Qin and Qiguang Chen contributed equally. This work was also sponsored by CCF-Baidu Open Fund. We are grateful for resources from the High Performance Computing Center of Central South University. Libo Qin is the corresponding author.

References

[Al-Rfou *et al.*, 2015] Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. Polyglot-NER: Massive multilingual named entity recognition. *Proc. of ICDM89*, 2015.

[Bastianelli *et al.*, 2020] Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. SLURP: A spoken language understanding resource package. In *Proc. of EMNLP*, 2020.

[Chen *et al.*, 2022a] Lisong Chen, Peilin Zhou, and Yuexian Zou. Joint multiple intent detection and slot filling via self-distillation. In *Proc. of ICASSP*, 2022.

[Chen *et al.*, 2022b] Lisung Chen, Nuo Chen, Yuexian Zou, Yong Wang, and Xinzhong Sun. A transformer-based threshold-free framework for multi-intent NLU. In *Proc. of COLING*, 2022.

[Cheng *et al.*, 2023a] Xuxin Cheng, Zhihong Zhu, Bowen Cao, Qichen Ye, and Yuexian Zou. MRRL: Modifying the reference via reinforcement learning for non-autoregressive joint multiple intent detection and slot filling. In *Proc. of EMNLP Findings*, 2023.

[Cheng *et al.*, 2023b] Xuxin Cheng, Zhihong Zhu, Wanshi Xu, Yaowei Li, Hongxiang Li, and Yuexian Zou. Accelerating multiple intent detection and slot filling via targeted knowledge distillation. In *Proc. of EMNLP Findings*, 2023.

[Coucke *et al.*, 2018] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*, 2018.

[Derczynski *et al.*, 2017] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proc. of WNUT*, 2017.

[Fetahu *et al.*, 2021] Besnik Fetahu, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. Gazetteer enhanced named entity recognition for code-mixed web queries. In *Proc. of SIGIR*, 2021.

[FitzGerald *et al.*, 2023] Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proc. of the ACL*, 2023.

[Gangadharaiah and Narayanaswamy, 2019] Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. Joint multiple intent detection and slot labeling for goal-oriented dialog. In *Proc. of NAACL*, 2019.

[Goo *et al.*, 2018] Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. Slot-gated modeling for joint slot filling and intent prediction. In *Proc. of NAACL*, 2018.

[Gururangan *et al.*, 2020] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proc. of ACL*, 2020.

[He *et al.*, 2023] Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *Proc. of ICLR*, 2023.

[Hemphill *et al.*, 1990] Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 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*, 1990.

[Liang *et al.*, 2020] Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proc. of EMNLP*, 2020.

[Ling and Weld, 2012] Xiao Ling and Daniel Weld. Fine-grained entity recognition. In *Proc. of AAAI*, 2012.

[Liu *et al.*, 2013] Jingjing Liu, Panupong Pasupat, Scott Cyphers, and Jim Glass. Asgard: A portable architecture for multilingual dialogue systems. In *Proc. of ICASSP*, 2013.

[Liu *et al.*, 2021] Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. *Benchmarking Natural Language Understanding Services for Building Conversational Agents*. 2021.

[Loshchilov and Hutter, 2019] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. of ICLR*, 2019.

[Pan *et al.*, 2017] Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. Cross-lingual name tagging and linking for 282 languages. In *Proc. of ACL*, 2017.

[Pan *et al.*, 2023] Wenbo Pan, Qiguang Chen, Xiao Xu, Wanxiang Che, and Libo Qin. A preliminary evaluation of chatgpt for zero-shot dialogue understanding. *arXiv preprint arXiv:2304.04256*, 2023.

[Pham *et al.*, 2023] Thinh Pham, Chi Tran, and Dat Quoc Nguyen. MISCA: A joint model for multiple intent detection and slot filling with intent-slot co-attention. In *Proc. of EMNLP Findings*, 2023.

[Pradhan *et al.*, 2013] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. Towards

robust linguistic analysis using OntoNotes. In *Proc. of CoNLL*, 2013.

[Qin *et al.*, 2019] Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. A stack-propagation framework with token-level intent detection for spoken language understanding. In *Proc. of EMNLP*, 2019.

[Qin *et al.*, 2020] Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. AGIF: An adaptive graph-interactive framework for joint multiple intent detection and slot filling. In *Proc. of EMNLP Findings*, 2020.

[Qin *et al.*, 2021a] Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu. A co-interactive transformer for joint slot filling and intent detection. In *Proc. of ICASSP*, 2021.

[Qin *et al.*, 2021b] Libo Qin, Fuxuan Wei, Tianbao Xie, Xiao Xu, Wanxiang Che, and Ting Liu. GL-GIN: Fast and accurate non-autoregressive model for joint multiple intent detection and slot filling. In *Proc. of ACL*, 2021.

[Qin *et al.*, 2021c] Libo Qin, Tianbao Xie, Wanxiang Che, and Ting Liu. A survey on spoken language understanding: Recent advances and new frontiers. In Zhi-Hua Zhou, editor, *Proc. of the IJCAI*, pages 4577–4584. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Survey Track.

[Qin *et al.*, 2023] Libo Qin, Qiguang Chen, Xiao Xu, Yunlong Feng, and Wanxiang Che. OpenSLU: A unified, modularized, and extensible toolkit for spoken language understanding. In *Proc. of ACL Demo*, 2023.

[Ritter *et al.*, 2011] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *Proc. of EMNLP*, 2011.

[Sang and De Meulder, 2003] Erik F Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proc. of AACL*, 2003.

[Schuster *et al.*, 2019] Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proc. of NAACL*, 2019.

[Song *et al.*, 2022a] Feifan Song, Lianzhe Huang, and Houfeng Wang. A unified framework for multi-intent spoken language understanding with prompting. *arXiv preprint arXiv:2210.03337*, 2022.

[Song *et al.*, 2022b] Mengxiao Song, Bowen Yu, Li Quangang, Wang Yubin, Tingwen Liu, and Hongbo Xu. Enhancing joint multiple intent detection and slot filling with global intent-slot co-occurrence. In *Proc. of EMNLP*, 2022.

[Sowański and Janicki, 2020] Marcin Sowański and Artur Janicki. Leyzer: A dataset for multilingual virtual assistants. In *International Conference on Text, Speech, and Dialogue*, 2020.

[Tedeschi and Navigli, 2022] Simone Tedeschi and Roberto Navigli. MultiNERD: A multilingual, multi-genre and fine-grained dataset for named entity recognition (and disambiguation). In *Proc. of ACL Findings*, 2022.

[Tedeschi *et al.*, 2021] Simone Tedeschi, Valentino Maiorca, Niccolò Campolungo, Francesco Cecconi, and Roberto Navigli. WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. In *Proc. of EMNLP Findings*, 2021.

[Tur and De Mori, 2011] Gokhan Tur and Renato De Mori. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[van der Goot *et al.*, 2021] Rob van der Goot, Ibrahim Sharaf, Aizhan Imankulova, Ahmet Üstün, Marija Stepanovic, Alan Ramponi, Siti Oryza Khairunnisa, Mamoru Komachi, and Barbara Plank. From masked-language modeling to translation: Non-English auxiliary tasks improve zero-shot spoken language understanding. In *Proc. of NAACL*, 2021.

[Wan *et al.*, 2023] Xue Wan, Wensheng Zhang, Mengxing Huang, Siling Feng, and Yuanyuan Wu. A unified approach to nested and non-nested slots for spoken language understanding. *Electronics*, 2023.

[Weischedel and Brunstein, 2005] Ralph Weischedel and Ada Brunstein. Bbn pronoun coreference and entity type corpus. *Linguistic Data Consortium, Philadelphia*, 2005.

[Wu *et al.*, 2022] Yangjun Wu, Han Wang, Dongxiang Zhang, Gang Chen, and Hao Zhang. Incorporating instructional prompts into a unified generative framework for joint multiple intent detection and slot filling. In *Proc. of COLING*, 2022.

[Xing and Tsang, 2022] Bowen Xing and Ivor Tsang. Co-guiding net: Achieving mutual guidances between multiple intent detection and slot filling via heterogeneous semantics-label graphs. In *Proc. of EMNLP*, 2022.

[Young *et al.*, 2013] Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. Pomdp-based statistical spoken dialog systems: A review. *Proc. of IEEE*, 2013.

[Zeldes, 2017] Amir Zeldes. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 2017.

[Zhu *et al.*, 2023] Zhihong Zhu, Weiyuan Xu, Xuxin Cheng, Tengtao Song, and Yuexian Zou. A dynamic graph interactive framework with label-semantic injection for spoken language understanding. In *Proc. of ICASSP*, 2023.