Low-Hanging Fruit: Knowledge Distillation from Noisy Teachers for Open Domain Spoken Language Understanding

Cheng Chen[1,3,4], Bowen Xing[5,6], and Ivor W. Tsang[2,3,4(✉)]

[1] Australian Artificial Intelligence Institute (AAII), University of Technology, Sydney, Australia
Cheng.chen-16@student.uts.edu.au
[2] School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore
[3] CFAR, Agency for Science, Technology and Research (A*STAR), Singapore, Singapore
ivor_tsang@cfar.a-star.edu.sg
[4] IHPC, Agency for Science, Technology and Research (A*STAR), Singapore, Singapore
[5] Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing 100083, China
[6] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

Abstract. Spoken Language Understanding (SLU) plays an integral role in dialogue systems. However, conventional SLU relies heavily on manually annotated datasets, which are impractical for open-domain SLU, given the wide variety of topics that must be considered. As the dataset grows exponentially, significant costs are inevitably incurred in achieving open-domain SLU. The Noisy Teacher and Consistently Guiding Student (NTCG) Paradigm is proposed to address these challenges. The objective is first to develop a prompt that effectively extracts valuable knowledge from large language models (LLMs), which can occasionally generate inconsistent and random responses, acting as 'noisy teachers.' This refined knowledge is then imparted to the downstream task model to improve performance further. To this end, we introduce an Incremental Progress Prompting Scheme (IPPS) under the NTCG that employs prompting techniques to generate more reliable annotations for unlabelled OD-SLU data, thereby fostering "Consistently Guiding Students". Initially, IPPS aims to solve the straightforward intent prediction task in OD-SLU using self-ranked prompting, enhancing LLMs precision using similar examples from a small, clean set as contextual hints for a given query. Additionally, the Intersection Sample Selection method is utilised to identify consistently predicted samples across different levels of randomness in ChatGPT, further improving its accuracy. The Consistent Intent Slot Prompting (CISP) method is proposed by exploiting the intent-to-slot correlation matrix to boost accuracy and precision for the more complex slot-filling task. Finally, the proposed Positively Fine-Tuned Scheme (PFTS) incorporates distilled knowledge from consistent

A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14944, pp. 107–125, 2024.
https://doi.org/10.1007/978-3-031-70359-1_7

samples via Label Consistency Regularisation to enhance downstream model performance. These strategies significantly improve intent detection and slot filling for prompt-based learning and downstream tasks.

Keywords: Open Domain Spoken Language Understanding (SLU) · Prompt based Task · Large Language Model (LLMs) · Knowledge Distillation

1 Introduction

Spoken Language Understanding (SLU) [48] hinges on the availability of high-quality annotations, which is crucial in a task-oriented system. It encompasses two sub-tasks: intent detection, a sentence-level classification task, and slot filling, a sequence labelling task. Over the past decade, significant advancements have been made in SLU. However, current methods often require extensive labelled data, which can be impractical for open-domain SLU (OD-SLU) [1,2,7,8] that needs to handle a wide range of topics. This necessity poses a great challenge for resource-limited groups such as small businesses or individuals seeking to customise dialogue systems for specific applications but lacking the financial and human resources to acquire a large amount of high-quality annotation. To mitigate this, an intuitive approach involves using Large Language Models (LLMs) to perform the annotation task. However, the responses generated by LLMs [32,39] often exhibit a degree of randomness and hallucinations, which leads to notable drawbacks. Such randomness and hallucinations can pose challenges, particularly when high-quality annotations are vital for open-domain spoken language learning tasks. Subsequently, the central question addressed in this paper is: ***'How can we extract valuable knowledge from a "Noisy Teacher" (ChatGPT) to train a Consistently-Guiding Student Model for Open-Domain Spoken Language Understanding (OD-SLU)?'***. To achieve this objective, we propose the Noisy Teacher and Consistently Guiding Student (NTCG) framework, which consists of two parts. The first part is the Incremental Progress Prompting Scheme (IPPS), and the second is the Positively Fine-Tuned Scheme (PFTS). The IPPS initially focuses on intent detection, which is the more straightforward task of OD-SLU, using self-ranked prompting for the noisy teacher. Specifically, we focus on mitigating the randomness in the "noisy teacher" of the intent detection task by using self-ranked prompting (see Sect. 4.1), inspired by [6]. This involves selecting an example as contextual hints most similar to a query from a small, clean sample and then feeding these into ChatGPT to generate the intent prediction. Our motivation stems from the observation that previous strategies have predominantly emphasized enriching the context, either by providing step-by-step explanations [43] or additional examples [42], to improve the quality of interactions with Large Language Models (LLMs). However, we believe that the effectiveness of prompting should not solely depend on the quantity or quality of the examples but also the semantic textual similarity between the Query (Q) and the example. Given the more

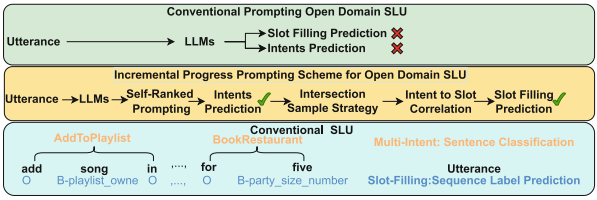

Fig. 1. Open Domain SLU for Multi-Intent Detection and Slot Filling Tasks.

relevant responses obtained using self-ranked prompting, an intersection sample selection strategy is applied (see Sect. 4.1). This approach aims to improve the consistency and precision of responses from ChatGPT by selecting only those samples that produce the same prediction across different levels of randomness in ChatGPT. This method enables the noisy teacher to reevaluate and reflect on the answers it provides to the student. Similar research, like that in [42], proposes choosing multiple, diverse samples with correct answers through few-shot CoT, utilizing the answers with the most consistent predictions as the final response. In the more challenging slot-filling task, one key challenge faced is the large number of slot label classes, which can range from 74 (MixSNIPS) to 121 (MixATIS), compared to intent detection, which typically involves 7 (MixSNIPS) to 18 (MixATIS) intent classes. This is a constrained generation task, meaning it involves querying the knowledge of LLMs from start to end, a process that has been shown to perform quite poorly by LLMs such as ChatGPT and the Bert Model [19]. Therefore, naively prompting large language models (LLMs) to provide the correct slot label for each word can lead to catastrophic results. To address this issue, we propose Consistent Intent Slot Prompting, which exploits the intent-to-slot correlation matrix to tackle the slot-filling task. Specifically, we use an intent-to-slot correlation matrix to reduce the original candidate set of slot label classes for consistent samples. As the intents of these consistent samples are better defined, the truncated slot labels for the corresponding intents and utterances become more useful. This substantially relieves the burden on the LLMs. While many prompting-based learning methods [3,3,11,25,29,46,49] have been studied to address some issues posed by noisy teachers, they fail to progressively solve sequential tasks cooperatively to address the challenges each task faces.

To achieve the "Consistently Guiding Student" model, we propose a Positively Fine-Tuned Paradigm (PFT). Moreover, we observe that the LLMs-generated predictions exhibit noisy multi-partial label supervision for the student model task. Based on this observation, we propose a positive fine-tuned contrastive loss for the student model, specifically designed for the LLM-generated prediction label set. The objective is to ensure that intra-class embeddings within the same class are brought closer together while those from different classes are pushed further apart. Subsequently, a more distinctive representation is learned, improving accuracy and robustness. Both methods form the basis of a "Noisy

Teacher and Consistently Guiding Student" paradiagm (see Sect. 4). The main contributions of this paper are:

- We introduce the "Noisy Teacher and Consistently-Guiding Student" learning framework for the intent and slot-filling tasks of OD-SLU.
- We propose incremental progress prompting to achieve more robust distillation for the intent and slot-filling tasks of OD-SLU.
- We demonstrate that leveraging semantically similar textual information between queries and examples in an unlabelled OD-SLU dataset can significantly improve both the subset accuracy and the overall accuracy of ChatGPT during prompting, especially when limited supervision is provided.
- We propose Consistent Intent Slot Inference Prompting, exploiting relevant intent-to-slot correlations matrix to improve the slot-filling task of OD-SLU.
- We propose a positively fine-tuned paradigm for addressing LLMs-generated noise multi-partial label learning for downstream tasks in intent detection.
- We construct a ChatGPT-generated predicted label set for the OD-SLU task based on MixATIS [16,35] and MixSNIPS [9,35].

2 Related Work

The challenge of multi-intent classification for spoken language understanding is initially addressed by [23]. Subsequently, [13] employs slot labels to tackle the multi-intent task. After that, [34] introduce auto-regressive modelling for multi-intent classification and slot filling, utilising graph attention networks [41]. An enhanced non-auto-regressive GAT model was later proposed, which improved the integration between the predicted intents and the hidden states of the slots. More recently, [44] developed both slot-to-intent and intent-to-slot graph neural networks, inspired by [41], enabling each network to guide the other in refining multi-intent and slot classifications. Recently, [6] proposes a self-teaching prompting approach that utilises Large Language Models (LLMs) to progressively generate and refine annotations for multi-intent datasets by learning from consistent samples. Knowledge distillation can be categorised into three types [14]: Response-based, where the student model learns from the teacher model's outputs [18,20]; Feature-based, where the student model is trained using features from the teacher model [17,36,47]; and Relation-based [28,33,38,40]. With the advent of larger LLMs like ChatGPT, the teacher model's role has evolved into a knowledgeable yet noisy source. Our work focuses not only on knowledge distillation but also on reducing randomness by identifying consistent and precise patterns in the outputs for the student model. Our positively fine-tuned contrastive loss function 8 is extended based on supervised contrastive learning [15,21] without using true label information. As our task involves multi-label classification, we have adapted the loss to suit multi-label scenarios. Previous studies [34,37] have explored multi-label learning with contrastive loss. However, our loss is the first to address noisy multi-partial label learning [26], a newly discovered branch of fine-grained partial label learning [5].

3 Problem Setting

We define the utterance space as $X \subseteq \mathbb{R}^d$, where d represents the dimensionality. Let $\mathcal{Y} = [k]$, with $[k] = \{1, 2, 3, ..., k\}$, denote the label space for the true label candidate set, where $k > 2$ indicates the number of classes. In this setting, for each instance x_i, the true label set is denoted by Y_i, where $Y_i \subseteq [k]$ and $Y_i \neq \emptyset$. This set Y_i contains the true labels associated with x_i. The set of all possible label combinations, excluding the empty and full sets, is denoted as $\mathcal{C}$. Formally, $\mathcal{C} = 2^{[k]} \setminus \{\emptyset, \mathcal{Y}\}$, where $2^{[k]}$ represents the power set of $\mathcal{Y}$. The size of this set is $|\mathcal{C}| = 2^k - 2$. For each instance x_i, the observed candidate label set $\overrightarrow{\boldsymbol{Y}}_i \in \mathcal{C}$ is the output of ChatGPT. This set $\overrightarrow{\boldsymbol{Y}}_i$ may include a partial or full subset of the true labels in Y_i and potentially false positive labels. The false positive labels F_i are denoted as $F_i \subseteq \{1, 2, ..., k\} \backslash Y_i$. Overall, a sample distribution with the predicted label sets generated by ChatGPT as $D_t = \{(x_1, \overrightarrow{\boldsymbol{Y}}_{t_1}), (x_2, \overrightarrow{\boldsymbol{Y}}_{t_2}), ..., (x_n, \overrightarrow{\boldsymbol{Y}}_{t_n})\}$ is given. Each tuple in D_t contains an instance x_i and its corresponding predicted label set $\overrightarrow{\boldsymbol{Y}}_i$, representing the noisy multi-partial label set.

4 Noise Teacher and Consistently Guiding Student Paradigm

The learning objective of the Noise Teacher and Consistently Guiding Student Paradigm is to enhance LLMs's response accuracy and improve the student model's robustness. The first part of the paradigm consists of refining LLMs's responses with self-ranked prompting and intersection sample selection strategies. The second component explains how to exploit refined knowledge of LLMs in the student model.

4.1 Incremental Progress Prompting Scheme for Intent and Slot Filling Distillation

Self-ranked Prompting for Intent. When given a query, our self-ranked prompting method selects the most similar and consistent ones with the query to assist the LLMs in predicting more accurately. The details of our proposed self-ranked prompt are illustrated in *Fig.* 2. Initially, an unlabelled dataset distribution $D_X = \{x_1, x_2, ..., x_n\}$ over the input space is provided. Additionally, a small, clean dataset is also given for prompt-based tasks. The clean data distribution is defined as $D_{AL} = \{(a_1, l_1), (a_2, l_2), ..., (a_s, l_s)\}$ and we have named it as hints, where s represents the total number of clean samples. Given D_X and D_{AL}, ChatGPT, denoted as G_t, is used to generate the predicted label sets $\overrightarrow{\boldsymbol{Y}}_t$ using our proposed self-ranked prompting. Here, t is the temperature parameter that controls the level of randomness in ChatGPT's label predictions for each input sample $x \in X$. We denote a_{TR} and m_{TR} as a top-ranked similar example (a selected utterance and its intent) selected from D_{AL} corresponding to each

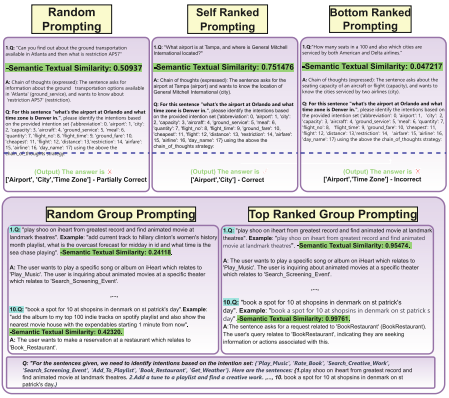

Fig. 2. Self-Ranked Prompting: The table shows the Self-Ranked Prompting strategy and prompting formatting. The semantic textual similarity is represented by the cosine similarity between $Query(x)$ and the example, denoted as a hint. A represents the corresponding intents and context information for the example. Our prompting studies include random prompting (baseline), top-ranking prompting, and bottom-ranking prompting. The score reflects the textual similarity between Q and hint. The group prompting format is at the bottom, and the single prompting format is at the top.

x_i, according to equation (3) and Fig. 2. Specifically, for each x_i, the predicted label set $\overrightarrow{\boldsymbol{Y}}_{t_i}$ is generated as follows:

$$G_t(x_i, a_{\mathrm{TR}}, m_{\mathrm{TR}}) = \overrightarrow{\boldsymbol{Y}}_{t_i}, \tag{1}$$

where,

$$a_{\mathrm{TR}}, m_{\mathrm{TR}} = \underset{(a_i, m_i) \in D_{\mathrm{AL}}}{\arg\max} \ \mathrm{ESC}(a_i, x). \tag{2}$$

This holds for all $i \in [N] = \{1, 2, 3, \ldots, N\}$, where N is the total number of training samples and for all $t \in \{0.1, 0.3, 0.5, 0.7\}$. We denote (a_{TR}, k_{TR}) as a hint and its intent corresponding to each query x for the ChatGPT. The embedding similarity score (ESC) is defined as follows:

$$\text{Embedding Similarity Score (ESC)} = \frac{W(\mathrm{a}) \cdot W(\mathrm{Query(x)})}{\|W(\mathrm{a})\| \|W(\mathrm{Query(x)})\|}, \tag{3}$$

where W represents the *word2vec* model [31], which is applied to obtain the lower-dimensional embeddings of both the hints and queries. The model is trained on sentences-only datasets. The ESC evaluates the word embedding similarity between the Query (x) and utterance a of the D_{AL}.

Intersection Sample Selection Strategy. Consequently, three sample distributions with varying temperature parameters t are obtained, denoted as:

$$D_t = \{(x_i, \overrightarrow{\boldsymbol{Y}}_{t_i}) | x_i \in X\}, \ \forall t \in \{0.1, 0.5, 0.3, 0.7\}. \tag{4}$$

The selection criterion for the consistent distribution is:

$$D_{consistent} = \{(x_{e_i}, \overrightarrow{\boldsymbol{Y}}_{e_i}) | x_{e_i} \in X \text{ and } \overrightarrow{\boldsymbol{Y}}_{0.1_i} = \overrightarrow{\boldsymbol{Y}}_{0.3_i} = \overrightarrow{\boldsymbol{Y}}_{0.5_i} = \overrightarrow{\boldsymbol{Y}}_{0.7_i}\}. \quad (5)$$

This criterion considers only those samples with the same predicted label sets across all three temperature-based ChatGPT configurations G_t.

Consistent Intent Slot Prompting for Open Domain Slot Filling Task. The slot-filling task in open-domain SLU can be challenging to solve, even with LLMs, due to the lengthy input and its corresponding slot label classes, which are often long and require the slot labels to be filled in a sequence corresponding to each word of the input utterance. Figure 1 illustrates the differences between the intent and slot-filling tasks. In our approach, we have further leveraged the obtained consistent dataset $D_{consistent}$ by proposing Consistent Intent-Slot Prompting (CISP) to exploit the intent-to-slot correlations matrix in addressing the more challenging slot-filling task of open-domain spoken language understanding (OD-SLU). For each intent prediction set $\overrightarrow{\boldsymbol{Y}}_{e_i}$ of a consistent sample x_{e_i}, each intent in the prediction set is associated with a set of slot labels. For instance, given the intent to slot correlation, we know that intent label *AddToPlaylist* is associated with the slot labels *playlist*, *entity_name*, and *artist*. We will only include these slot labels in the slot class candidate set for the x_{e_i} slot labels. Given that the original size of the full slot class candidate set is large, applying our approach can significantly reduce the size of the slot class candidate set. This reduction alleviates the burden on large language models (LLMs) and improves slot-filling predictions. Figure 4 details the intent-to-slot correlations matrix for both datasets. The correlation between intent and slot is inductive, meaning it can be acquired heuristically (Table 4).

4.2 Positively Fine-Tuned Paradigm

Consistently-Guiding Student Via Consistent Samples. In this subsection, we aim to improve the robustness of the student model by leveraging the samples of distribution $D_{consistent}$ to address challenges that arise from the noisy multi-partial label type supervision generated by the ChatGPT. Within a batch, let $i \in I \equiv \{1 \dots N\}$ be the index of a sample drawn i.i.d from the distribution $D_{t=0.3}$, and let $j \in P \equiv \{1 \dots N_+\}$ be the index of a consistent sample drawn i.i.d from the distribution $D_{consistent}$. Here, $N_+ = |D_{consistent}|$ and $N = |D_t|$. The consistent sample originates from the same source sample as the sample but includes these samples with the same prediction label set across all three randomness configurations of ChatGPT. More specifically, given an sample x_i, it is only considered as the consistent sample x_j if the condition $G_{0.1}(x_i) = G_{0.5}(x_i) = G_{0.7}(x_i)$ is satisfied. We define $A(i) \equiv I \backslash \{i\}$. The index i is denoted as an anchor. However, we are not using index i as an anchor. Instead, we used index j and named it an equivalence anchor in our revised loss. We define $A(j) \equiv I \backslash \{j\}$, the set of all indices in the batch excluding index j.

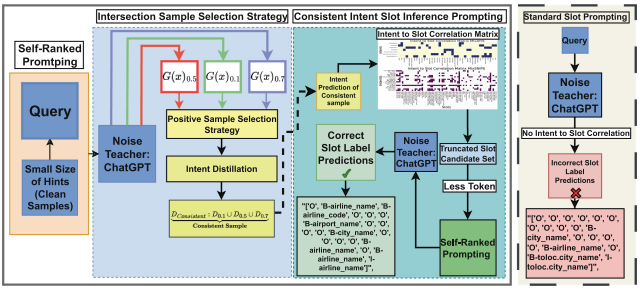

Fig. 3. Our proposed Incremental Progress Prompting Scheme for knowledge distillation from a noisy teacher for open-domain spoken language understanding involves three steps. The first part uses self-ranked prompting to enable LLMs to generate outputs with less noise. Subsequently, an intersection sample selection strategy is employed to obtain consistent samples. Given the consistent sample and estimated intent-slot correlation, we truncate the candidate slot label classes to enable LLMs to generate more refined responses for the slot-filling task.

$P \subseteq I$ denotes the indices of consistent samples in the batch. $P(j) \equiv P \backslash \{j\}$, the index j is equivalence anchor (Table 2).

Label-Wise Embedding Regularisation. The key issue of noisy multi-partial label generated prediction is the chance of a complete set of true labels not being guaranteed to exist in the candidate label set, contrary to the multi-partial label setting where true labels are always present. Thus, exploiting the candidate label set of consistent samples with a significantly higher accuracy rate helps the loss function to identify more precise positive samples.

$$C_+(j) = \{c | c \in P(j), i \in I : M_{ci} > 1\}, \tag{6}$$

the $C_+(j)$ is designed to identify samples with sufficiently higher label similarity to the equivalence anchor to be considered a positive sample. The M is defined as follows:

$$M_{ic} = \begin{cases} 1 & \text{if } \overrightarrow{\boldsymbol{Y}}_i^{\,T} \cdot \overrightarrow{\boldsymbol{Y}}_c > 1 \\ 0 & \text{otherwise} \end{cases}. \tag{7}$$

From equation (7), every embedding of equivalence anchor will be pulled together with embeddings of batch samples from the same class that met the criterion $M_{ci} > 1$. The equivalence anchor embedding is pulled apart with embeddings of batch samples with other classes that do not meet the criterion $M_{ci} > 1$. The intra and inter-class correlation matrix M is designed to learn more distinctive representations by exploiting as much information as possible from the labels. The $\overrightarrow{\boldsymbol{Y}}_c$ is the predicted label vector of the c^{th} equivalence anchor. $\overrightarrow{\boldsymbol{Y}}_i$ denote as the predicted label vector of the i^{th} sample in the batch. The entry of M_{ic} is greater than 1 if there is an overlap of the c^{th} predicted label vector and the i^{th}

sample's predicted label set, and 0 if there are no shared labels. This approach addresses the issues of noisy multi-partial labels by focusing on the most reliable and consistent label relationships.

Equivalent Anchors Using Reliable Feature Representation. Next, we train a decoder f and a projection head g by leveraging the embeddings of consistent samples. We refer to consistent samples as equivalent anchors. The embedding $z = g(x)$ represents the encoder's lower embedding, the backbone model's L2-normalised final hidden state layer. It transforms the instance x into a dense vector representation $z \in \mathbb{R}^d$ for a sample x. Given this, we have defined the positive fine-tuned contrastive loss function as follows:

$$\mathcal{L}(f(x), \tau, C, A) = -\sum_{j \in P} \frac{1}{|C_+(j)|} \sum_{c \in C_+(j)} \log \frac{\exp({z_j}^\top z_c/\tau)}{\sum_{a \in A(j)} \exp(z_j^\top z_a/\tau)}, \tag{8}$$

Our work utilises equivalence anchors, which are consistent samples selected using the intersection sample selection strategy. These anchors are employed uniquely in our contrastive loss function, differing from methods in previous works [12,22,30,37]. By contrasting these equivalence anchors, which have demonstrated much higher accuracy (as shown in Tables 1 and 2), against other samples in the batch, our loss function enables the model to learn more precise and distinct representations despite the noise in multi-partial labels. This approach ensures that the embeddings of equivalence anchors from identical categories are brought closer together, aligning with those of other samples that exhibit similar annotation characteristics while distancing them from samples of different classes.

Label Consistency Regularisation Using Intersection Sample Prior. To further mitigate the harmfulness caused by the false positive labels, inspired by maxi-margin assumption [4], we have proposed intersection sample prior label-aware consistency regularisation $\mathcal{L}_{\text{ISPL}}$ Eq. (9) to utilise the prior label distribution as regularisation, adjusting the logits to help the classifier down weight the frequently occurring false positive class and up weight the less frequently occurring positive class during training process. For each sample in the batch and each class, we compute the prior modified uniform matrix as:

$$T_{ij} = \overrightarrow{\boldsymbol{Y}}_{0.3+ij} \cdot m_{ij}, \quad \forall i \in \{1, \ldots, n\},\ j \in \{1, \ldots, k\},$$

where $\overrightarrow{\boldsymbol{Y}}_{+ij}$ is an predicted label set of instance x_i i.i.d draw from the consistent distribution $D_{consistent}$. The m_j is denoted as:

$$m_{i,j} = exp(\alpha \cdot \max(\lambda_{i,j} \cdot N_j)) \quad , \forall j \in \{1, ..., k\},$$

where k is the number of classes, j denotes the index for each class, λ is the class proportion, and it is estimated from consistent distribution. The α is the

hyperparameter. The T_{ij} is the $i-th$ sample's target value for the j-th class. Subsequently, we adjust the original logits Q_{ij} with the predefined T_{ij}, which is defined as $Q_{ij} = f(x)$. The modified logits $V_{m_{ij}}$ is defined as $V_{m_{ij}} = Q_{ij} - \mathrm{T}_{ij}$, where m_{ij} is the i-th sample's logit for the j-th class. Intersection Sample Prior Label Consistency Regularisation is defined as follows:

$$\mathcal{L}_{\mathrm{ISPL}} = -\frac{1}{K}\sum_{j=1}^{K}\left[\overrightarrow{\boldsymbol{Y}}_{0.3_j}\log(\sigma(V_{m_j})) + (1-\overrightarrow{\boldsymbol{Y}}_{0.3_j})\log(1-\sigma(V_{m_j}))\right], \tag{9}$$

where $\sigma(V_{m_j})$ is the sigmoid function applied to the logit for class j. The $\overrightarrow{\boldsymbol{Y}}_{0.3_{ij}}$ is the predicted label set for the x_i. The $V_{m_{ij}}$ is the modified logit of Q_{ij}. The λ_1 is a hyper-parameter. To encapsulate, the final learning objective for the downstream task manifests as:

$$\mathcal{L} = (1-\lambda_2)(\mathcal{L}_{\mathrm{ISPL}} + \lambda 1\mathcal{L}(f(x), \tau, C, A)) + \mathbb{I}\cdot(\lambda_2)\mathcal{L}_{\mathrm{BCE}}, \tag{10}$$

where $\mathbb{I}$(Intent and Slot Task) is the indication function. In our downstream experiments, we consider both LLMs-generated intent prediction and LLMs-generated intent prediction with clean slot-filling tasks. $\mathbb{I}$ indicates that in the former case, we will not consider $\mathcal{L}_{\mathrm{BCE}}$, whereas if both intent and slot tasks are considered, $\mathcal{L}_{\mathrm{BCE}}$ is included. Our downstream task focuses on intent classification, but our proposed method has also improved the slot-filling task. The λ_2 becomes 1 when $\mathbb{I}$(Intent and Slot Task) is zero.

5 Experiment

Datasets. The experiments are implemented on two open-source multi-intent datasets, MixATIS and MixSNIPS. MixATIS [16,35] includes 13,162 utterances for training, 756 for validation and 828 for testing. MixSNIPS includes [9,35] 39,776, 2,198, and 2,199 utterances for training, validation and testing datasets.

Prompting Experiment Setting. We use the ChatGPT 3.5 [32] to help us generate the annotations for our downstream task on the BERT [10], XLnet [45] and ROBERTA [27] models. We allow 65 tokens for each query. We have set the temperature parameter t at $0.1, 0.3, 0.5, 0.7$ for our prompting task. Our entire experiment is conducted using the group prompting approach. Group prompting refers to asking multiple questions in each query. For each query, we have given 5 and 10 utterances to ChatGPT on dataset MixATIS and MIXSNIPS, respectively. We have initially randomly selected 74 samples with true intents for our predefined D_{AL}^{ATIS} from the MIXATIS dataset and 400 samples with true intents for our predefined D_{AL}^{SNIPS} from the MIXSNIPS dataset.

Table 1. Comparison of Self-Ranked Prompting and Chain of Thought Prompting in the Intent Prediction Task for MixSNIPS and MixATIS

t	0.1	0.3	0.5	0.7	Average
MixATIS - Chain of Thought Prompting					
Accuracy Ratio	0.31	0.30	0.30	0.29	0.30
Subset Ratio	0.52	0.52	0.52	0.50	0.51
MixATIS - Self-Ranked Prompting					
Accuracy Ratio	0.37	0.38	0.38	0.34	0.37
Subset Ratio	0.57	0.58	0.58	0.55	0.57
MixSNIPS - Chain of Thought Prompting					
Matching Ratio	0.40	0.41	0.44	0.46	0.43
Subset Ratio	0.45	0.46	0.48	0.50	0.47
MixSNIPS - Self-Ranked Prompting					
Accuracy Ratio	0.66	0.68	0.68	0.69	0.68
Subset Ratio	0.75	0.76	0.77	0.77	0.76

Evaluation Metrics for Prompting. This metric measures the exact matching rate between the predicted label set and the true label set. It can be calculated as:

$$\text{Accuracy Ratio} = \frac{\text{Number of correctly predicted labels}}{\text{Total number of samples}} \tag{11}$$

This metric measures the ratio of the predicted label set that includes the true label set. It can be calculated as:

$$\text{Subset Ratio} = \frac{\text{Number of predicted label sets that includes true labels}}{\text{Total number of samples}} \tag{12}$$

These metrics help evaluate the performance of our proposed prompting method in terms of intent prediction accuracy and the subset intent prediction (Inclusion of true labels in the predicted sets)

Table 2. Consistent Distribution Generation Via Intersection Sample Selection: Results for MixATIS and MIXSNIPS Datasets with Intersection Sample Selection. The 3468 of 13162 is the sample size for consistent sample distribution MixATIS. The 23186 of 39776 is the sample size for consistent sample distribution MIXSNIPS.

Dataset	Metric	$D_{consistent}$	% of the Dataset Left
MixATIS	Accuracy Ratio	**59.02%**	26.34%
MixATIS	Subset Ratio	63.41%	26.34%
MIXSNIPS	Accuracy Ratio	**80.78%**	58.29%
MIXSNIPS	Subset Ratio	87.74%	58.29%

Evaluation Metrics for Downstream Task. We have used the following metrics to evaluate the performance of downstream tasks. Precision(P) $= \frac{TP}{TP+FP}$, Recall(R) $= \frac{TP}{TP+FN}$, F1 Score $= 2 \times \frac{P \times R}{P+R}$, Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$. TP denotes the number of true positive samples. TN means the number of true negative samples. FP is defined as the number of false positives. FN is the number of false negative samples. P is precision and R is recall.

Experimental Details for Downstream Task. The hyperparameter α is defined as $\alpha = \log(0.75) + 0.25$. The threshold τ is set to 0.0001 for the MixATIS and MixSNIP datasets. For tasks involving LLM-generated intent and clean slots, τ is set to 0.0001 for MixATIS and 0.00001 for MixSNIP, with the total number of epochs set to 35 and 50, respectively. The best performance on the validation dataset set is chosen for reporting the final results. The λ_2=0.01. We have used the Bert model [10], XLnet [45], and Roberta [27], which is robustly optimised BERT pretraining approach as backbones for our intention classification downstream task. We used a binary cross-entropy logit loss function in the baseline based on the BERT, Roberta, and XLnet models. The learning rate is set at 0.001, whereas the rate of dropouts in the network is set at 0.1, and the batch size is set at 16 for the MixATIS and 32 for the MixSNIPS. The total epoch is set at 30. We use *Adam* [24] as our optimiser. The λ_1 is 0.1 for the MixSNIPS and 0.01 for the MixATIS (Tables 5 and 6).

Table 3. Comparison of Random and Top Self-Ranked Prompting methods on the MixSnips and MixATIS datasets. The Group Random and Top Ranking are shown in Fig. 2.

Dataset	Prompting	Accuracy Ratio	Subset Ratio
MixSNIPS	Group Random Ranking	64.75 ± 2.65%	70.75 ± 12.37%
	Group Top Ranking	**74.19** ± 3.86%	**81.18** ± 8.49%
MixATIS	Group Random Ranking	37.00 ± 10.54%	56.00 ± 1.00%
	Group Top Ranking	**39.75** ± 0.50%	**58.75** ± 4.03%

Consistent Distribution Generation Via Intersection Sample Selection. After we obtained predicted label sets for both the MixATIS and MixSNIPS datasets, using randomness parameter of $\{0.1, 0.3, 0.5, 0.7\}$ applying self-ranked prompting, the "Intersection Sample Selection" method was applied to obtain a consistent distribution. The consistent distribution demonstrated superior matching and subset ratio performance compared to the self-ranked prompting-only approach. The performance improvement was more significant in the MixSNIPS dataset, where the Accuracy Ratio increased from 0.6869 (Confidence 0.7) to 0.8078 (Intersection Sample Selection), and the Subset Ratio rose

from 0.7715 (Confidence 0.7) to 0.8774 (Intersection Sample Selection). There was also a noticeable improvement in the MixATIS dataset; the Accuracy Ratio went up from 0.3767 (Confidence 0.3) to 0.5615 (Intersection), and the Subset Ratio climbed from 0.5764 (Confidence 0.7) to 0.6784 (Intersection). Overall, our experimental results have proven that "Intersection Sample Selection" is an effective strategy for enhancing both matching and subset ratios in both datasets (Table 2).

Table 4. A Comparison between Consistent Intent Slot Prompting (CISP) and Standard Slot Prompting for Slot Filling Task on MixATIS and MixSNIP.

MixSNIP	Our (CISP)	BaseLine
Total Exact Match	**3.85%**	0.57%
Average F1 Score	**24%**	7%
MixATIS	**Our (CISP)**	**BaseLine**
Total Exact Match%/Number	**6.49%**	3.15%
Average F1 Score	**35%**	26%

Consistent Intent Slot Prompting for Slot Filling Experiment. This section compares Consistent Intent Slot Prompting (CISP) and Standard Slot Prompting for the Slot Filling Task on MixATIS and MixSNIP. Our method has improved LLMs' matching rate and F1 score for the slot filling task by 2.28 % and 17% on MixSNIP and 3.34% and 9% on MixATIS, respectively. Even though the exact matching and F1 score improvements are significant when employing our Consistent Intent Slot Prompting, they are still insufficient. Therefore, we have not used the ChatGPT-generated slot labels for downstream tasks.

Downstream Task "MixATIS" and "MixATIS" on the Slot Filling and Intent Experimental Results. We have used the LLMs-generated intent and clean slot to evaluate our proposed ISPL+PFTS loss function. Table 7 compares the performance of BERT models using Our Proposed Method and BCE loss on the MixATIS and MixSNIP datasets. For the MixATIS dataset, the table indicates that our ISPL+PFTS loss function outperforms the baseline BCE Loss in terms of intent accuracy **(+4.45%)** and slot F1 score (+**6.98**). Likewise, for the MixSNIP dataset, an improvement is shown when using Our method (ISPL+PFTS) compared to the baseline in terms of intent accuracy (+**1.24**%) and slot F1 score (+**1.31**%). Our proposed loss function for the downstream task is focused on intent classification, but our proposed method has also shown improvements in the slot-filling task.

Table 5. Comparison and Improvement of MixATIS with Different Models and PFTS Loss.

Dataset	Model	Loss Function	Intent Acc.	Precision	Recall	F1-Score
MixATIS						
	BERT	BCE Loss	51.96 ± 2.00%	71.90 ± 1.23%	**71.33** ± 2.43%	70.11 ± 2.06%
	BERT	(ISPL+PFTS)	**54.88** ±3.01%	**73.19** ±0.51%	71.52 ±1.58%	**70.95** ±1.19%
	Roberta	BCE Loss	51.14 ± 1.17%	71.79 ± 1.55%	70.65 ± 1.78%	69.63 ± 0.89%
	Roberta	(BCE+ISPL+PFTS)	**53.48** ±0.78%	**72.53** ±0.60%	**71.65** ±0.69%	**70.86** ±0.52%
MixSNIPS						
% Training samples						
5%	BERT	ISPL + PFTS	**70.72** ± 1.14%	90.91 ± 0.88%	90.76 ± 0.52%	90.77 ± 0.29%
5%	BERT	BCE	69.31 ± 1.20%	91.41 ± 0.20%	88.15 ± 0.080%	89.26 ± 0.06%
10%	BERT	ISPL + PFTS	**79.44** ± 0.56%	95.57 ± 0.79%	90.88 ± 0.71%	93.03 ± 0.19%
10%	BERT	BCE	76.23 ± 1.32%	95.34 ± 1.5%	89.15 ± 0.84%	91.98 ± 0.47%
50%	BERT	ISPL + PFTS	**85.73** ± 0.55%	96.50 ± 0.45%	94.76 ± 1.57%	95.03 ± 0.05%
50%	BERT	BCE	83.33 ± 1.06%	96.94 ± 0.57%	91.98 ± 0.69%	94.25 ± 0.29%
100%	BERT	ISPL + PFTS	**87.55** ± 0.61%	96.66 ± 0.19%	94.55 ± 0.37%	95.50 ± 0.20%
100%	BERT	BCE	85.76 ± 1.15%	97.00 ± 0.20%	93.20 ± 0.67%	94.94 ± 0.39%
100%	Robert	ISPL + PFTS	**88.93** ± 0.12%	96.75 ± 0.23%	95.40 ± 0.18%	96.01 ± 0.16%
100%	Robert	BCE	86.56 ± 0.44%	97.22 ± 0.42%	93.42 ± 0.62%	95.16 ± 0.32%
100%	X-Lnet	ISPL + PFTS	**88.55** ± 0.61%	96.66 ± 0.44%	94.75 ± 0.28%	95.57 ± 0.18%
100%	X-Lnet	BCE	85.79 ± 1.09%	97.06 ± 0.51%	93.22 ± 0.316%	94.98 ± 0.10%

Table 6. Comparison and Improvement of MixATIS and MixSNIP Datasets for Intent and Slot Filling Using Our Method (ISPL+PFTS) Vs the Baseline BCE Loss Model.

Dataset	Model	Loss Function	Intent Accuracy	Slot F1 Score
MixATIS	BERT	BCE Loss	30.37 ±1.45%	14.35 ± 0.39%
	BERT	ISPL+PFTS	**34.82** ±3.76%	**21.33**±0.58%
MixSNIP	BERT	BCE Loss	72.26 ± 1.15%	14.90 ± 0.82%
	BERT	ISPL+PFTS	**73.50** ±1.91%	**16.21** ± 0.78%

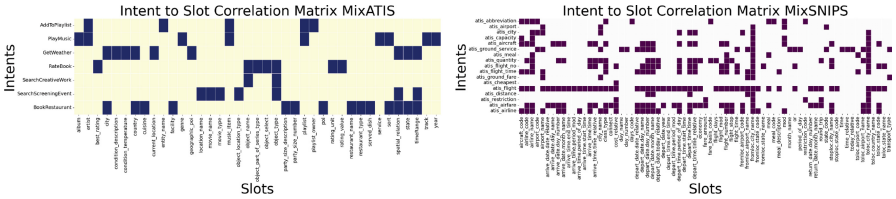

Fig. 4. Intents to Slots Correlation Matrices on MixSNIP and MixATIS

Downstream Task "MixATIS" Experimental Results. BERT Model: For the MixATIS dataset, we achieved a 2.92% improvement in intent accuracy, 1.29% improvement in precision, 0.19% improvement in Recall, 0.84% improvement in F1-Score compared to the standard BCE loss on the BERT model. All results are averaged over 6 random seeds. **Roberta Model:** For the MixATIS dataset, we achieved a 2.34% improvement in intent accuracy, 0.74% improvement in preci-

sion, 1.00% improvement in Recall, 1.23% improvement in F1-Score compared to the standard BCE loss on the Roberta model.

Downstream Task "MixSNIPS" Experimental Results. BERT Model: For the MixSNIP dataset, we achieved a 1.79% improvement in intent accuracy, −0.34% in precision, 1.35% improvement in Recall, 0.56%improvement in F1-Score compared to the standard BCE loss on the BERT model. All results are averaged over 6 random seeds. **Roberta Model:** Our ISPL + PFTS loss function has shown a notable improvement, with a 2.37% increase in accuracy compared to the baseline. **X-LNET Model:** Analogous to the Robert model, there is a significant 2.76% improvement in intent accuracy with our ISPL + PFTS loss function. All results are averaged over 6 random seeds.

Table 7. Comparison of Intent Accuracy Among BCE Loss Function, BCE + PFTS, and ISPL+PFTS Loss Functions. Diff 1=(ISPL+PFTS vs BCE+PFTS) and Diff 2= (ISPL+PFTS vs BCE)

Sample Size	BCE Loss	BCE + PFTS	ISPL+PFTS	Diff 1	Diff 2
100%	85.76 ± 1.15	86.31 ± 0.63	88.55 ± 0.61	+2.24	+2.79
50%	83.33 ± 1.06	84.11 ± 0.94	87.55 ± 0.61	+3.44	+4.22
10%	76.23 ± 1.32	77.33 ± 1.47	79.44 ± 0.56	+2.11	+3.21
5%	69.31 ± 1.20	70.35 ± 0.97	70.72 ± 1.14	+0.37	+1.41

Ablation Study. We have conducted an ablation study on MXISNIP to evaluate the effectiveness of the ISPL and PFTS loss functions. This study compared the standard BCE loss with the BCE+PFTS and ISPL+PFTS loss functions. The BCE loss combined with PFTS demonstrated improvements in intent accuracy of 1.04%, 1.1%, 0.78%, and 0.55% for 5%, 10%, 50%, and 100% of the training samples, respectively. In comparison, the ISPL+PFTS loss function achieved improvements in intent accuracy of 0.37%, 2.11%, 3.44%, and 2.24% for the 5%, 10%, 50%, and 100% of the training samples, respectively, when compared against the BCE + PFTS loss function. Notably, the ISPL+PFTS loss function showed improvements of 1.41%, 3.21%, 4.22%, and 2.79% for the 5%, 10%, 50%, and 100% sample sizes, respectively. Additionally, we have conducted an ablation study (See Table 8) on various scenarios using the single prompting method to evaluate the effectiveness of self-ranked prompting. It shows that having more examples per query helps improve both the accuracy and subset ratios. Additionally, increasing the size of the predefined D_{AL} helps increase the accuracy ratio, especially effective with the "only intents" example method. Contextual information is useful in improving the subset ratio. The study of single prompting is significant because, in some applications, users tend to ask only one query at a time. This is often the case with voice assistants or in medical-related inquiries.

Table 8. Experimental Results for the Prompting Task (Single Prompting) on Small Sample Sizes. The contextual information consists of a step-by-step explanation of the given query. The baseline refers to a prompting method that asks a question without displaying the selected example's contextual information or the intent of the selected example. 'Other methods' include prompting incorporating contextual information, both contextual information and intent or only intent.

15 positive examples per sentence	Baseline	Contextual information	Contextual information and intent	only intent
Subset Ratio	53.0%	58.82%	59.02%	63.37%
Accuracy Ratio	12.4%	11.56%	18.55%	18.82%
Recall Ratio	23.4%	19.65%	31.44%	29.70%
small pool (15 samples)	Baseline	Contextual information	Contextual Information and Intent	only intents
Subset Ratio	53.0%	60.04%	57.87%	56.05%
Accuracy Ratio	12.4%	13.81%	13.79%	17.59%
Recall Ratio	23.4%	23.00%	23.83%	31.38%
large pool (157 samples)	Baseline	Contextual information	Contextual Information and Intent	only intents
Subset Ratio	53.0%	54.50%	55.55%	57.12%
Accuracy Ratio	12.4%	8.81%	10.12%	15.43%
Recall Ratio	23.4%	16.16%	18.22%	27.01%

6 Discussion

This paper proposes a "noisy teacher and Consistently Guiding student" learning paradigm for the open-domain spoken language understanding (SLU) task. On the LLMs side, incremental progress prompting scheme is proposed to solve the easier intent task of OD-SLU and then tackle the more challenging slot-filling task. We propose self-ranked prompting and intersection sample selection for intent task distillation to derive consistent samples, enhancing relevance and consistency for downstream tasks. Furthermore, we exploit the consistent samples and intent-to-slot correlations matrix to facilitate slot-filling prediction using LLMs. Lastly, for the "consistently guiding student" model, we introduce a positively fine-tuned contrastive loss and intersection sample prior label consistency regularisation to further improve intent classification performance. By applying our paradigm, future research could focus on utilising LLMs to detect new and multilingual intent.

Acknowledgement. We would like to express our profound gratitude to all the anonymous reviewers and the area chair for their invaluable comments.

References

1. Aghaebrahimian, A., Jurčíček, F.: Constraint-based open-domain question answering using knowledge graph search. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2016. LNCS (LNAI), vol. 9924, pp. 28–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45510-5_4
2. Aghaebrahimian, A., Jurcícek, F.: Open-domain factoid question answering via knowledge graph search. In: Proceedings of the Workshop on Human-Computer Question Answering, pp. 22–28 (2016)

3. Brown, T., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
4. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
5. Chen, C., Lyu, Y., Tsang, I.W.: Adversary-aware partial label learning with label distillation. arXiv preprint arXiv:2304.00498 (2023)
6. Chen, C., Tsang, I.: Self-teaching prompting for multi-intent learning with limited supervision. In: The Second Tiny Papers Track at ICLR 2024 (2024). https://openreview.net/forum?id=DeoamI1BFh
7. Chen, D., Yih, W.t.: Open-domain question answering. In: Savary, A., Zhang, Y. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, pp. 34–37. Association for Computational Linguistics (2020)
8. Cheng, H., Shen, Y., Liu, X., He, P., Chen, W., Gao, J.: UnitedQA: a hybrid approach for open domain question answering. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) 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), pp. 3080–3090. Association for Computational Linguistics (2021)
9. Coucke, A., et al.: Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. arXiv preprint arXiv:1805.10190 (2018)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
11. Diao, S., Wang, P., Lin, Y., Zhang, T.: Active prompting with chain-of-thought for large language models (2023)
12. Fei, Y., Nie, P., Meng, Z., Wattenhofer, R., Sachan, M.: Beyond prompting: making pre-trained language models better zero-shot learners by clustering representations. arXiv preprint arXiv:2210.16637 (2022)
13. Gangadharaiah, R., Narayanaswamy, B.: Joint multiple intent detection and slot labeling for goal-oriented dialog. 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), pp. 564–569. Association for Computational Linguistics, Minneapolis, Minnesota (2019)
14. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: a survey. Int. J. Comput. Vision **129**, 1789–1819 (2021)
15. Gunel, B., Du, J., Conneau, A., Stoyanov, V.: Supervised contrastive learning for pre-trained language model fine-tuning. arXiv preprint arXiv:2011.01403 (2020)
16. Hemphill, C.T., Godfrey, J.J., Doddington, G.R.: 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)
17. Heo, B., Lee, M., Yun, S., Choi, J.Y.: Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 3779–3787 (2019)
18. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
19. Hu, E.J., et al.: Amortizing intractable inference in large language models. arXiv preprint arXiv:2310.04363 (2023)

20. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3588–3597 (2018)
21. Khosla, P., et al.: Supervised contrastive learning. In: Advances in Neural Information Processing Systems, vol. 33, 18661–18673 (2020)
22. Khosla, P., et al.: Supervised contrastive learning. CoRR abs/2004.11362 (2020)
23. Kim, B., Ryu, S., Lee, G.G.: Two-stage multi-intent detection for spoken language understanding. Multimedia Tools Appl. **76**, 11377–11390 (2017)
24. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
25. Liu, J., et al.: Generated knowledge prompting for commonsense reasoning. arXiv preprint arXiv:2110.08387 (2021)
26. Liu, W., Wang, H., Shen, X., Tsang, I.W.: The emerging trends of multi-label learning. IEEE Trans. Pattern Anal. Mach. Intell. **44**(11), 7955–7974 (2021)
27. Liu, Y., et al.: Roberta: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
28. Liu, Y., et al.: Knowledge distillation via instance relationship graph. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7096–7104 (2019)
29. Liu, Z., Yu, X., Fang, Y., Zhang, X.: Graphprompt: unifying pre-training and downstream tasks for graph neural networks. In: Proceedings of the ACM Web Conference 2023 (2023)
30. Malkinski, M., Mandziuk, J.: Multi-label contrastive learning for abstract visual reasoning. CoRR **abs/2012.01944** (2020). https://arxiv.org/abs/2012.01944
31. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
32. OpenAI: Gpt-4 technical report (2023)
33. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3967–3976 (2019)
34. Qin, L., Wei, F., Xie, T., Xu, X., Che, W., Liu, T.: GL-GIN: fast and accurate non-autoregressive model for joint multiple intent detection and slot filling. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) 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), pp. 178–188. Association for Computational Linguistics, Online (2021)
35. Qin, L., Xu, X., Che, W., Liu, T.: AGIF: an adaptive graph-interactive framework for joint multiple intent detection and slot filling. arXiv preprint arXiv:2004.10087 (2020)
36. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: hints for thin deep nets. arXiv preprint arXiv:1412.6550 (2014)
37. Su, X., Wang, R., Dai, X.: Contrastive learning-enhanced nearest neighbor mechanism for multi-label text classification. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 672–679. Association for Computational Linguistics, Dublin, Ireland (2022)
38. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. arXiv preprint arXiv:1910.10699 (2019)
39. Touvron, H., et al.: Llama: open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)

40. Tung, F., Mori, G.: Similarity-preserving knowledge distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1365–1374 (2019)
41. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
42. Wang, X., et al.: Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171 (2022)
43. Wei, J., et al.: Chain-of-thought prompting elicits reasoning in large language models. In: Advances in Neural Information Processing Systems, vol. 35, pp. 24824–24837 (2022)
44. Xing, B., Tsang, I.W.: Co-guiding net: aarXiv preprint arXiv:2210.10375 (2022)
45. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: XLNet: generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
46. Yao, S., et al.: React: synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629 (2022)
47. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: fast optimization, network minimization and transfer learning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7130–7138 (2017)
48. Young, S., Gašić, M., Thomson, B., Williams, J.D.: Pomdp-based statistical spoken dialog systems: A review. Proc. IEEE **101**(5), 1160–1179 (2013)
49. Zhou, Y., et al.: Large language models are human-level prompt engineers. arXiv preprint arXiv:2211.01910 (2022)