Speech-based Slot Filling using Large Language Models

Anonymous ACL submission

Abstract

Recently, advancements in large language models (LLMs) have shown an unprecedented ability across various language tasks. This paper investigates the potential application of LLMs to slot filling with noisy ASR transcriptions, via both in-context learning and task-specific fine-tuning. Dedicated prompt designs and fine-tuning approaches are proposed to improve the robustness of LLMs for slot filling with noisy ASR transcriptions. Moreover, a linearised knowledge injection (LKI) scheme is also proposed to integrate dynamic external knowledge into LLMs. Experiments were performed on SLURP to quantify the performance of LLMs, including GPT-3.5-turbo, GPT-4, LLaMA-13B, LLaMA-2-13B and Vicuna-13B (v1.1 and v1.5) with different ASR error rates. The use of the proposed fine-tuning approach together with the LKI scheme for Vicuna-13B-v1.5 achieved 8.1% and 21.5% absolute SLU-F1 improvements compared to the strong Flan-T5-base baseline system on the limited data setup and the zero-shot setup respectively.

1 Introduction

Slot filling, as an important sub-task of spoken language understanding (SLU), is a crucial component in conversational AI such as spoken dialogue systems. It requires the extraction and understanding of pertinent information in the user's speech query. Accurate extraction of slot values from the query speech is indispensable for accurate response generation and is challenging with limited annotated data and noisy ASR transcriptions. In particular, domain-specific named entities that are crucial to accurate information extraction, usually have high error rates with a generic ASR system. Although this problem can be mitigated by training systems on data in the target domain, it can be expensive to construct dedicated large-scale training data for a specific SLU task as it requires an extensive labelling effort and domain expertise (Hou et al., 2020; Liu et al., 2020; Henderson and Vulić, 2021). This data sparsity problem can be addressed by transfer learning with pre-trained language models (PLMs) (Du et al., 2021; Fuisz et al., 2022), especially with large language models (LLM).

Recent advancements in LLMs, such as GPT-4 (Ouyang et al., 2022; OpenAI, 2023) and the LLaMA series (Touvron et al., 2023a,b), have been shown to exhibit human-level reasoning ability for natural language tasks even without task-specific fine-tuning of the model parameters, which is known as the *emergence* of LLMs. This is usually achieved by conditioning the model generation process on a prompt containing examples or a task description, referred to as in-context learning. Although effective, studies have also shown that LLMs struggle with accurate fine-grained content extraction such as slot-filling (Heck et al., 2023; Zhang et al., 2023b; Pan et al., 2023; Shen et al., 2023), and tend to overly extrapolate beyond the examples and task descriptions in the prompt, especially when evaluated using text-based quantitative metrics. This necessitates the use of dynamic contextual knowledge to guide and confine the generation (Omar et al., 2023; Peng et al., 2023), as well as efficient task-specific fine-tuning with limited data (Zhang et al., 2023a). Moreover, as slot-filling relies on the quality of ASR transcriptions (Seo et al., 2022; Raju et al., 2022; Rao et al., 2021; Sun et al., 2023b), it is essential to improve the performance of LLMs with noisy ASR outputs.

This paper aims to apply LLMs for slot filling with different ASR error rates under limited data scenarios. A dedicated prompt design specifically targeting LLMs is proposed featuring task descriptions, linearised dynamic external knowledge and multiple alternative ASR hypotheses. The task description, together with in-context examples, provides the basis of in-context learning using LLMs. In addition to the one-best hypothesis, the proposed prompt design can incorporate multiple hypotheses in the form of an N-best list for a single utterance to improve the model's robustness to ASR errors. Moreover, by leveraging a pre-defined knowledge base (KB) (Sun et al., 2023b), dynamic knowledge found in the N-best list is also linearised into text and used in the prompt to provide the necessary constraint to guide the language generation[1].

Experiments were performed using SLURP data, where particular attention was paid to the performance of in-context learning, few-shot learning and zero-shot learning for unseen slot types. Several LLMs were investigated, including GPT-3.5-turbo and GPT-4 for in-context learning and LLaMA and Vicuna models as widely-used open-source models for fine-tuning. Different sizes of Whisper ASR models were adopted as a group of generic off-the-shelf models to provide transcriptions with different ASR error rates. The experimental results showed that instruction-tuned LLMs outperformed the baseline GPT-2 model and Flan-T5-base model in both in-context learning and limited data fine-tuning setups. Moreover, the fine-tuned Vicuna-13B-v1.5 model outperformed the Flan-T5-base by 22.4% in absolute SLU-F1 under the zero-shot setup, which was also close to the performance on seen slot types. The main contributions of this paper can be summarised as follows.

- The performance on slot filling is determined using SLURP data for a range of widely used LLMs, including GPT-3.5-turbo, GPT-4, LLaMA and Vicuna.
- A prompt design and data-efficient noise-robust fine-tuning approach for slot filling using LLMs with noisy ASR transcriptions is provided.
- A linearised knowledge injection (LKI) scheme is proposed that incorporates contextual knowledge derived using N-best ASR hypotheses into the prompt for LLMs.

2 Related Work

2.1 LLMs

LLMs refer to the type of LMs with billions of model parameters and trained on vast amounts of data. GPT-3 (175 billion parameters) (Brown et al., 2020) and PaLM (540 billion parameters) (Chowdhery et al., 2022) were early examples of LLMs, significantly outperforming their predecessors such as BERT (Devlin et al., 2019) (330 million parameters) and GPT-2 (Radford et al., 2019) (1.5 billion parameters). One of the most prominent applications of LLMs is ChatGPT, which was built from GPT-3.5 via reinforcement learning with human feedback (RLHF) (Christiano et al., 2017) and was adapted for chat applications. Later, a larger scale LLM, GPT-4 (OpenAI, 2023), was built to further support the visual modality. While the capability of LLMs continued to expand with ever-growing model sizes, "smaller" LLMs (still with tens of billions of parameters) such as LLaMA (Touvron et al., 2023a) were released and achieved a better balance between the performance and the computing resource required. Since LLaMA is open-source, many versions of LLaMA such as Vicuna (Zheng et al., 2023) have been developed with different conversation-based fine-tuning schemes.

With scaled-up model and training data sizes, LLMs have demonstrated a superior ability for solving assorted complex tasks over their predecessors, which are known as "emergent abilities" (Wei et al., 2022). One of the key capabilities of LLMs is in-context learning, which has enabled LLMs to perform specific tasks by providing task descriptions or examples without explicitly updating model parameters. Early research explored in-context learning by providing the schema of the ontology in a spoken dialogue system. Specifically, in (Chen et al., 2020), domain-slot relations from the dialogue ontology were encoded using a graph neural network (GNN) to guide the system. Later, (Hu et al., 2022) proposed an in-context learning framework by summarising the dialogue history into a short description, together with the schema description, as the context. More recently, (Ouyang et al., 2022) proposed a more powerful LLM trained via RLHF that performed response generation via in-context learning.

2.2 Slot Filling with Limited Data

As slot-filling often requires domain expertise for labelling which makes it very expensive, data efficiency has been a crucial research topic. Fine-tuning PLMs on a relatively small-scale task-specific dataset for slot-filling has been one of the main themes in this area. In particular, compared to sequence tagging using pre-trained bi-directional encoder systems (Henderson and Vulić, 2021), by formulating slot-filling as a sequence generation task, the power of language generation of LMs, such as GPT-2 (Budzianowski and Vulić, 2019) or T5 (Raffel et al., 2019; Wu et al., 2022; Lin et al.,

[1]Code and prompt template will be available at [URL]

2021b), can be further exploited. These models achieved few-shot learning by only presenting a limited number of data samples, and it was also discovered that providing task descriptions in the context helps few-shot learning. Meanwhile, integrating prior knowledge about slots, such as possible values of each slot, also showed improvements in performance (Lin et al., 2021a; Wang et al., 2022), especially with limited data(Sun et al., 2023b).

LLMs have enabled unprecedented slot-filling performance without task-specific fine-tuning by only presenting the LLM with a task description and several examples. The LLM will generate the desired slot and value pairs in the standard LM fashion (Pan et al., 2023; Shen et al., 2023; Heck et al., 2023). Structured contextual knowledge can be linearised into plain text as a part of the context to further improve in-context learning ability (Xie et al., 2022). Despite this ability and the performance achieved, the quantitatively measured slot-filling performance across a wide range of domains is still far from that of state-of-the-art fine-tuned systems (Pan et al., 2023; Shen et al., 2023).

3 Methodology

3.1 Prompt Design for Slot Filling

The prompt design is shown in Table 1, which comprises five major parts: *task definition*, *in-context examples*, linearised knowledge injection (*LKI*), *query* and *additional query with LKI*.

The task definition gives a brief definition for each slot type. It is necessary to encourage the generated response to focus on the slot types that are needed. The in-context examples provide further context to improve the performance. This part provides few-shot examples which comprise pairs of utterances and desired outputs in the prompt. Although in-context examples provide effective guidance on what to generate, the number of samples in the prompt is always limited by the context length. The LKI and the additional query with LKI are two parts that appear in pairs when an external KB is used. As LLMs tend to extrapolate and create an excessive number of slot-value pairs, these two KB-related parts are used to provide a strong prior that confines the generation of certain slots to the allowed set of possible values. The main query prompts LLMs with the question, followed by utterances from which slot values are extracted. The basic form of the prompt is the concatenation of the task definition and the query, followed by the utterance, with the other three parts being optional.

The complete pipeline for slot filling with LLMs taking speech as input is shown in Fig. 1. Starting with the input speech, a Whisper model is used to transcribe the speech into an N-best list. The prompt is then constructed according to the design, with an additional global prompt for systems with instruction tuning. The top K hypotheses, where $K = 1$ by default, are used as the final part of the prompt. The complete prompt is given to the LLM to generate a text sequence that completes the assistant response using a search algorithm as shown in Eqn (1).

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{W}^P), \qquad (1)$$

where $\mathbf{W}$ is the generated token sequence and $\mathbf{W}^P$ is the prompt token sequence. Note that there is no stochasticity in the generation process as the task is to extract the exact information from an utterance for quantitative measurements.

The LKI part of the prompt improves the robustness of LLMs by constraining the generation with a pre-defined KB which is especially important with noisy ASR transcriptions. The external KB contains all possible named entities for each slot type. Then, for each utterance, entities that can be found in the N-best list via string matching are selected together with their slot types, as shown in the example in Fig. 1. The selected slot values are linearised into a text format to be used as part of the prompt. The use of LKI not only guides the generation process so that sensible values constrained by the KB are generated but also makes further use of ASR alternatives. For example, the name “pawel" in the utterance has different substitutions (e.g. “power"), and when using the top hypothesis for slot filling, the LLM is unable to fill this slot with the correct value. However, the name is correctly recognised in the 4th-best hypothesis and provided the entry in LKI. With that extra information in the prompt, the LLM has a much better chance of correctly performing slot filling for that entity.

3.2 Task-specific Finetuning

Task-specific finetuning is particularly useful when a small amount of training data is available to leverage the few-shot learning ability of LLMs. The low-rank adaptation (LoRA) (Hu et al., 2021) finetuning method is used in this paper.

LoRA approximates the necessary update to a full-rank parameter matrix using a low-rank matrix, which can be decomposed into the multipli-

Table 1: Prompt design for slot filling using LLMs, including task definition, in-context examples, linearised knowledge injection (LKI) scheme, query and the extra constraint for LKI only used for in-context learning. The example shows $K = 1$ in this table.

Prompt parts	Prompt template
Task definition	Consider the following list of slot types provided to you in JSON format: {"`slot A`": "definition of slot A", "`slot B`": "definition of slot B", ...}
In-context examples	For example, given "`utterance_1`" you should extract {"`slot A`": "`value A`"}, given "`utterance_2`" you should extract {"`slot B`": "`value B`"}, etc.
LKI	First, possible values for some slots are provided: {"`slot A`": "`value A`", ...} Slot values not in the KB are not likely to be the correct value
Query	Now consider the following sentence(s) containing one or more of the above slot types. Extract slots belonging to that list and their values in JSON format. i.e. {"slot type": "slot value"}, or {} if no slots found
Additional instruction	If, for a slot in KB, you can not find a value in KB, select most likely values from KB instead

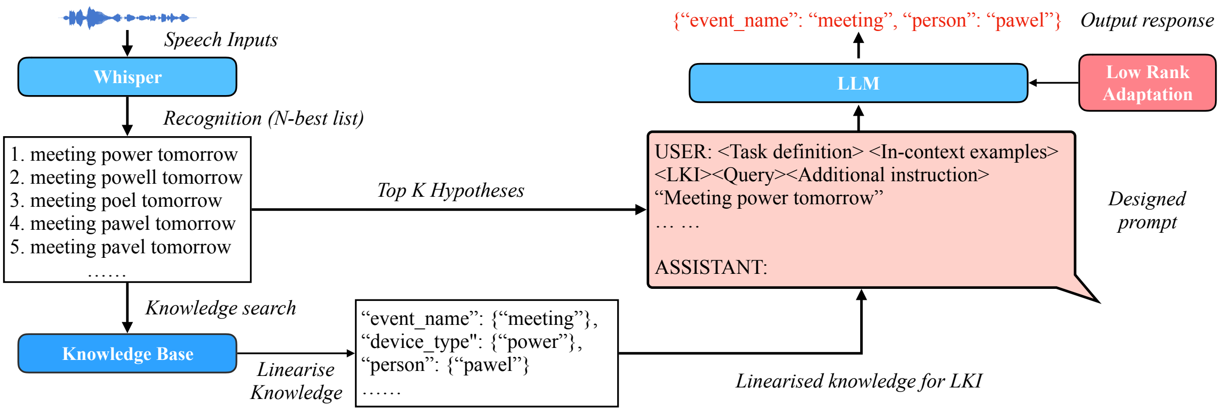

Figure 1: Diagram illustrating the slot-filling pipeline using Whisper and LLM. The user prompt comprises parts introduced in Table 1, followed by the top one hypothesis in the N-best list as an example. Low-rank adaptation (LoRA) is used for fine-tuning open-source LLMs. Note that the specific format of the roles in the prompt, e.g. ASSISTANT, is dependent on the specific LLM.

cation of two low-rank matrices. Specifically, for a pre-trained matrix $\mathbf{W}_0 \in \mathbb{R}^{m\times n}$ from a specific attention block, its update during finetuning can be constrained in the form shown in Eqn. (2).

$$\mathbf{W}_0 + \Delta\mathbf{W} = \mathbf{W}_0 + \mathbf{AB} \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{m\times r}$ and $\mathbf{B} \in \mathbb{R}^{r\times n}$ contain trainable parameters with the pre-trained parameters, $\mathbf{W}_0$, fixed, and $r \leq \min(m, n)$. When setting r to be a value much smaller than the model dimension, e.g. 8, the total number of parameters to be learned during fine-tuning is much reduced, which both improves the training efficiency and avoids over-fitting with a large number of parameters, especially for LLMs. Following (Hu et al., 2021), LoRA is applied to the projection matrices of the self-attention layer in this paper.

LLMs are trained in the standard LM fashion with the cross-entropy (CE) loss applied to the generated output, i.e. the slot-value pairs in JSON format in Fig. 1. As the ASR system is an off-the-shelf Whisper model which is not fine-tuned in this paper, and, given that the training and validation set WER are similar, the N-best list derived from the training audio can be used to improve the robustness of LLM to the noisy transcriptions. Specifically, instead of feeding in the reference transcription, the top K most likely hypotheses can instead be concatenated into one single string and used in the prompt during fine-tuning so that the model learns to robustly extract information from multiple hypotheses. In addition, LKI can be used during fine-tuning so that the model is aware of the knowledge provided and learns to use it. Random distracting entities can also be added to the LKI part to avoid the model being over-confident about the accuracy of the provided knowledge.

4 Experimental Setup

4.1 Data

SLURP (Bastianelli et al., 2020) is a collection of

25k single-turn user interactions with a home assistant, annotated with scenarios, actions and entities. Experiments were performed in two data setups. First, the official training, validation and test split following (Bastianelli et al., 2020) was used where synthesised audio files were also included. Note that subsets of the training set were selected, with e.g. 2000 samples corresponding to 8% of the full data, to demonstrate the limited data scenarios. In addition, a simulated zero-shot setup (Sun et al., 2023b) was used. In training, all utterances containing entities of five randomly selected 'unseen' slots were held out, and the held-out set containing 2000 utterances was used for testing.

The KB was organised as a simple dictionary for SLURP, where the keys were slot types and the values were lists of possible named entities for that type. It was created by collecting named entities that appeared in the entire SLURP data for each slot type, including train, validation and test sets, as a simulation of a real-world task environment. The average size of these lists was 106: the largest list was `person` which contained 872 entities, and the smallest list was `transport_agency`, which only contained 2 entities.

4.2 Models

Six different LLMs were selected for evaluation in this paper: GPT-3.5, GPT-4, LLaMA(-2)-13B and Vicuna-13B (v1.1 and v1.5). The first two models were chosen as the two most popular LLMs representing the state-of-the-art performance of in-context learning without parameter fine-tuning. The other models were chosen as widely used open-source models that allow task-specific fine-tuning. Each model is introduced below:

GPT-3.5 was used in ChatGPT, a chatbot application introduced by OpenAI. Although exact details about its architecture and training procedures were not published, according to OpenAI, it used a similar architecture to Ouyang et al. (2022) (175B parameters) and was trained with RLHF. The version released on March 1st, 2023 (GPT-3.5-turbo-0301) was used for the experiments.

GPT-4 went one step further from GPT-3.5 by having a larger scale (1050B parameters) and additionally accepting images as input, though only the text modality was explored in this paper. In the experiments, the version released on March 16th, 2023 (GPT-4-0316) with a maximum sequence length of 4096 was used.

LLaMA-13B, together with its improved version **LLaMA-2-13B**, is a series of LLMs with a Transformer decoder structure. The model contains 13B parameters and was pre-trained on one trillion tokens. It includes various techniques used in training other LLMs such as pre-normalisation (Zhang and Sennrich, 2019), the SwiGLU activation function (Shazeer, 2020), and rotary embeddings (Su et al., 2022).

Vicuna-13B (version 1.1) is a fine-tuned version of LLaMA which is adapted specifically to chat applications. It was trained on user-shared conversations with ChatGPT collected using a tool called ShareGPT. The model contained 13B parameters. The upgraded version, **Vicuna-13B-v1.5**, derived from the LLaMA-2 model was also used in this paper. The version supporting 16k tokens was used to represent the long context model.

As baselines and performance references, the GPT-2 model with 117 million parameters and the FLAN-T5 base model with 250 million parameters were employed that were fine-tuned on SLURP with all parameters[2]. Four different sizes of English Whisper models including tiny (39M), base (74M), small (244M) and medium (769M), were used as the off-the-shelf ASR models. The Whisper large model achieved very similar performance to the medium model and for efficiency, the medium model was used in the experiments. Both the reference and ASR transcriptions were normalised following the text normalisation scheme in (Radford et al., 2022), and were scored against a normalised SLURP label file[3].

4.3 Training and Inference Configurations

The LLaMA and Vicuna models were fine-tuned using LoRA with rank 8 which adapts 6.5M parameters, i.e. equivalent to 0.05% of the original model parameters. During training, the LKI part was organised by selecting entities that appeared in the reference transcription, and adding random distractors found in the ASR hypotheses of the same utterance. All systems were trained on a single A100 GPU which took 2 hours to fine-tune a 13B LLaMA model under the 2000 sample setup.

During inference, a greedy search algorithm was used for all LLMs until the end-of-sequence token was output. The result was then fed to a post-

[2] GPT-2 system already achieved the previous best SLU-F1 on cascaded system (Sun et al., 2023a), and Flan-T5-base achieved further improvements compared to GPT-2.

[3] The Influence of written form is analysed in Appendix B

Table 2: SLU-F1 (Precision/Recall) of LLMs with transcriptions from different Whisper ASR systems on SLURP test set. %WERs of Whisper systems are also provided. LLMs were evaluated using task descriptions and a one-shot example as the prompt. GPT-2 and Flan-T5 were fine-tuned on the full SLURP training data as a reference.

Whisper Model WER	Tiny (%) 33.4	Base (%) 26.4	Small (%) 19.8	Medium (%) 14.6	Reference (%) 0.0
GPT-3.5	34.7 (32.0/38.0)	36.7 (34.1/39.7)	38.6 (35.0/42.7)	40.8 (36.3/46.7)	46.5 (40.5/54.5)
GPT-4	41.3 (43.5/39.3)	42.3 (42.1/42.5)	45.2 (43.1/47.6)	47.0 (43.7/50.9)	53.6 (49.4/58.6)
Vicuna-13B	7.6 (4.5/20.1)	7.7 (5.0/17.0)	8.1 (5.2/17.9)	8.9 (5.8/19.3)	16.5 (11.5/29.5)
Vicuna-13B-v1.5	16.6 (14.9/18.8)	17.8 (16.3/19.6)	20.7 (18.5/23.6)	19.9 (17.3/23.3)	22.4 (18.4/28.5)
GPT-2 Full data	54.9 (66.0/47.0)	58.4 (70.6/49.9)	64.0 (72.6/57.3)	65.2 (74.1/58.2)	84.8 (85.5/84.1)
Flan-T5-base Full data	56.4 (69.2/47.6)	59.1 (72.2/50.8)	64.6 (74.3/57.2)	66.5 (76.7/58.8)	86.3 (87.1/85.5)

processing stage using regular expressions to extract slot-value pairs in JSON format from the output text. As LLMs may fail to follow the exact instructions, for any utterances that fail to extract valid JSON format text, an empty output would be given. The LKI part was extracted based on the N-best hypotheses from the corresponding Whisper model. Systems were evaluated using the SLU-F1 metric, which combines both word-level and character-level F1 scores to give partial credit to non-exact match predictions.

Table 3: SLU-F1 scores of LLMs using few-shot in-context learning on SLURP test set with the reference or Whisper medium model transcriptions. GPT-2 and Flan-T5 models were fine-tuned on the number of samples indicated. Each sample took 20 tokens on average. Vicuna-13B was unable to have 100 examples in the prompt due to the model's maximum context lengths.

Systems	N-samples	Medium (%)	Ref. (%)
GPT-3.5	0	40.1	45.9
GPT-4	0	46.4	53.0
Vicuna-13B	0	7.6	12.2
Vicuna-13B-v1.5	0	17.6	18.3
GPT-3.5	1	40.8	46.5
GPT-4	1	47.0	53.6
Vicuna-13B	1	8.9	16.5
Vicuna-13B-v1.5	1	19.9	22.4
GPT-3.5	10	42.3	46.8
GPT-4	10	49.9	55.9
Vicuna-13B	10	11.8	21.4
Vicuna-13B-v1.5	10	21.3	24.9
GPT-3.5	100	47.4	53.5
GPT-4	100	**57.4**	65.8
Vicuna-13B-v1.5	100	25.8	31.3
GPT-2*	100	33.5	38.9
GPT-2*	2000	52.2	65.2
Flan-T5-base*	2000	55.2	**68.9**

5 Results and Discussion

5.1 In-context learning with noisy ASR transcriptions

The performance of LLMs on slot filling via in-context learning with one-shot prompts is summarised in Table 2 using different Whisper models. Note that the LLaMA models without fine-tuning were not able to follow instructions and hence not able to perform slot filling. In general, LLMs yielded a higher recall rate than precision as they were not trained with the confined scenario setting and tended to pick up all possible fillers for each slot based on their knowledge. This problem became more severe when the definition of the slot type was rather abstract, such as `event_name` or `news_topic`, where LLMs tended to extract almost the entire utterance as the filler. The problem was less severe for slots requiring a single named entity to fill, e.g. `person`. For all systems, the degradation in recall was more than the degradation in precision when the ASR error rate increased, as important entities tended to be incorrectly recognised by the generic Whisper systems. Note that the degradation due to noisy transcriptions would probably be much smaller with an ASR system fine-tuned on SLURP audio with only a slightly lower overall WER (Sun et al., 2023a), as it is more likely to recognise domain-specific entities correctly.

Compared to the Vicuna-13B models containing 13B model parameters, GPT-3.5 and GPT-4 with much larger model sizes achieved much better results without any further parameter updates, which demonstrated the emergent ability of LLMs when the model size reached a certain level. GPT-4 achieved around a 7% increase in SLU-F1 compared to GPT-3.5 under different ASR conditions while yielding a better balance between precision and recall. While GPT-2 and Flan-T5-base with task-specific fine-tuning achieved much better results with reference transcriptions, the performance degradation of using noisy transcription is also much larger compared to the non-fine-tuned LLMs.

An important aspect of in-context learning is to offer examples so that LLM is able to learn from them. To investigate the effect of the number of in-context learning samples, the performance against the number of samples for different systems is shown in Table 3. As the number of samples in the prompt increased (except for the 100-sample case with Vicuna-13B which exceeded the maximum token length of the model), there was an increase in the SLU-F1 found in all LLMs both with reference and noisy ASR transcriptions, at a cost of increasing prompt sequence lengths. In particular, GPT-4 achieved a larger improvement than GPT-3.5 when more samples were included, indicating its superior capability in leveraging context information in the prompt. Notably, GPT-4 with 100 samples in-context learning performed much better than a GPT-2 model fine-tuned on the same number of samples and also surpassed the performance of a Flan-T5-base model fine-tuned on 2000 samples when using noisy transcription derived from the Whisper medium model. However, with 100 samples, the length of the prompt with 100 samples is close to 3000 while the 10-sample prompt which is only 1000. This significantly increased the inference cost with API calls.

5.2 Task-specific fine-tuning

Next, task-specific fine-tuning was applied to LLaMA-13B and Vicuna-13B v1.1 using LoRA on subsets of the SLURP training set for limited data scenarios. To begin with, SLU-F1 on the SLURP test set against different training set sizes was plotted as shown in Fig. 2. With 500 and 2000 training samples representing limited data scenarios (2% and 8% of the full training set respectively), the performance of LLaMA-13B and Vicuna-13B achieved much better performance than GPT-2. The performance of Vicuna-13B was slightly worse than LLaMA-13B. When more samples were included, LLMs fine-tuned with LoRA gradually lost their advantage over the fully fine-tuned LMs. Further improvements may require full model fine-tuning for LLMs which necessitates distributed model parameters across multiple GPUs. In the following experiments, the 2000-sample limited data setup was used where LLMs achieved better performance than the best in-context learning performance achieved by GPT-4. A similar trend was observed under Whisper medium model transcriptions (see Appendix A).

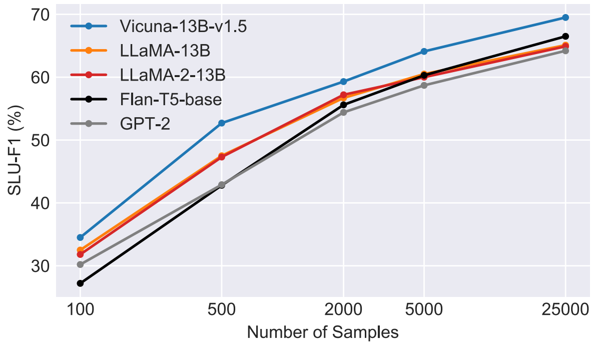

Figure 2: SLU-F1 on the standard SLURP test set using Whisper medium model transcriptions against the number of samples in the training set for fine-tuning five LMs. Note that 25000 represents the full training set.

Table 4: SLU-F1 (Precision/Recall) on SLURP test set using transcriptions from the Whisper medium model and linearised knowledge injection (LKI) in both in-context learning and fine-tuning (indicated with FT) setups. Fine-tuning was performed on the 2000-sample subset, and “13B" was omitted for clarity.

Systems	With LKI (%)	No LKI (%)
GPT-3.5	46.8 (36.1/66.4)	40.8 (36.3/46.7)
Vicuna	21.8 (15.3/37.4)	8.9 (5.8/19.3)
Vicuna-v1.5	33.0 (27.3/41.8)	19.9 (17.3/23.3)
GPT-2 FT	57.0 (65.8/50.2)	52.2 (61.8/45.0)
Flan-T5-base FT	59.5 (68.3/52.7)	55.2 (66.3/47.3)
LLaMA FT	62.6 (71.2/55.9)	56.7 (69.2/48.0)
Vicuna FT	61.3 (70.0/54.5)	53.4 (61.5/47.2)
LLaMA-2 FT	61.8 (65.4/58.5)	57.2 (62.4/53.8)
Vicuna-v1.5 FT	**63.8** (68.0/59.9)	**59.3** (65.3/54.3)

Linearised knowledge injection provided important external information about the application context. To this end, the effect of such knowledge injected using the proposed LKI approach was investigated as shown in Table 4 for both in-context learning and fine-tuning. As shown in Table 4, the main effect of LKI was to improve the recall as it mainly provided information for entities that were not in the top-1 hypothesis. LKI for GPT-3.5 had a rather limited effect, as while improving the recall, it decreased the precision as noise in the LKI prompt introduced many irrelevant slot values. The influence of LKI on Vicuna-13B under in-context learning almost doubled the SLU-F1, although the numbers were still much lower than GPT-3.5. When fine-tuned with LKI in the prompt, all LMs achieved larger improvements in the overall SLU-F1, as fine-tuned models had a much lower recall. Vicuna-13B-v1.5 and LLaMA-2 benefited the most from LKI as they could process long context better, and the best performance was achieved

Table 5: SLU-F1 (Precision/Recall) on SLURP test set with LLMs fine-tuned on noisy transcriptions from Whisper medium model. The training settings indicate the number of hypotheses concatenated and used in the prompt. The same number of hypotheses in training was used during inference. All LLMs were fine-tuned and fine-tuning was performed on the 2000-sample subset.

System	Training	SLU-F1 (%)
GPT-2	ref	52.2 (61.8 / 45.0)
GPT-2	10-best	39.0 (62.3 / 28.3)
Flan-T5-base	ref	55.2 (66.3 / 47.3)
Flan-T5-base	10-best	35.2 (59.2 / 25.1)
Vicuna-13B-v1.5	ref.	59.3 (65.3 / 54.3)
Vicuna-13B-v1.5	1-best	60.1 (69.2 / 53.2)
Vicuna-13B-v1.5	5-best	63.3 (66.1 / 60.7)
Vicuna-13B-v1.5	10-best	63.4 (65.9 / 60.9)
Vicuna-13B-v1.5	20-best	61.2 (64.3 / 58.4)
Vicuna-13B-v1.5	9-best + ref.	62.4 (66.9 / 56.1)
Vicuna-13B-v1.5 LKI	10-best	**65.6** (75.3 / 58.1)

by Vicuna-13B-v1.5. As a result, the proposed LKI approach proved to be an effective way for dynamic contextual knowledge integration in LLMs.

Fine-tuning with multiple hypotheses from the ASR system improves the robustness of LLMs to ASR errors for slot filling. The results of using different numbers of hypotheses are summarised in Table 5. As shown in the first part of Table 5, using the 10-best hypotheses in GPT-2 and Flan-T5-base confused the model and yielded much worse results with a severely degraded recall rate. On the other hand, using N-best hypotheses in the context during training improved the performance of LLMs, as demonstrated in the second part of Table 5. Although using references for training achieved better performance than the 1-best hypotheses, incorporating the references into N-best lists for training yielded worse performance as the model still relied on the reference presented in the N-best list.

The best SLU-F1 score was achieved by combining the proposed LKI with the use of noisy transcriptions for fine-tuning, yielding an overall **6.3**% absolute SLU-F1 increase compared to using the 1-best prompt and **8.1**% absolute increase compared to the strong Flan-T5-base baseline system. Although the knowledge was selected based on the N-best hypotheses, LKI was still informative when the N-best list was included in the context and achieved a 2.2% absolute increase in SLU-F1.

5.3 Generalisation to unseen slots

To further demonstrate the power of the proposed approaches, the generalisation of various LLMs to unseen slot types was studied using the simulated

Table 6: SLU-F1 (Precision/Recall) on SLURP zero-shot test set containing unseen slots. Slot-filling systems were finetuned using 2000 samples and the full training set. Whisper medium outputs were used.

System	2000 samples (%)	Full (%)
Flan-T5-base	25.2 (28.7 / 22.4)	25.1 (28.3 / 22.6)
+ LKI	36.5 (42.2 / 32.2)	32.4 (38.3 / 28.0)
LLaMA-2-13B	32.6 (42.7 / 26.3)	32.1 (40.7 / 26.5)
+ LKI	54.3 (66.7 / 45.7)	57.0 (70.5 / 47.8)
+ 10-best	56.5 (58.6 / 54.5)	58.2 (74.3 / 47.8)
Vicuna-13B-v1.5	37.0 (41.0 / 33.7)	36.5 (42.7 / 31.9)
+ LKI	57.0 (66.1 / 50.1)	61.6 (76.4 / 51.7)
+ 10-best	**58.0** (68.3 / 50.3)	**64.0** (80.5 / 53.1)

zero-shot setup. The SLU-F1 of LLaMA-2-13B and Vicuna-13B-v1.5 on the held-out test set are shown in Table 6 compared to Flan-T5-base.

As shown in Table 6, when trained on 2000 samples, Vicuna-13B-v1.5 achieved 12% absolute SLU-F1 improvement compared to the Flan-T5-base model, as LLMs were better at leveraging the task description part in the prompt design. In addition, such improvement was further increased to **21.5**% when LKI and N-best hypotheses were applied, indicating that LLMs were much better at using external knowledge provided in the context. However, providing further in-domain training samples did not provide as much improvements to the performance in this out-of-domain test set. It is also worthwhile highlighting that LLMs with the proposed approaches achieved a performance close to that on the standard test set even though the majority of slot types in the zero-shot test set have never appeared in training.

6 Conclusions

This paper investigated the performance of LLMs for slot filling with noisy transcriptions from speech. It quantified the performance of six different LLMs using ASR transcriptions from four different sizes of Whisper ASR models and proposed a dedicated prompt design, a noise-robust fine-tuning approach and a linearised knowledge integration (LKI) scheme. The proposed prompt design with few-shot examples enabled GPT-4 to outperform a GPT-2 model fine-tuned on 20 times more data. With fine-tuning, the Vicuna-13B-v1.5 model achieved an absolute 8.1% and 21.5% SLU-F1 increase using both the noise-robust fine-tuning and the LKI scheme compared to a strong fully fine-tuned Flan-T5-base model on few-shot and zero-shot scenarios respectively.

7 Limitation

The main limitation of using LKI and N-best hypotheses for training is the context length of LLMs. When sequence length becomes much longer by incorporating more examples or more hypotheses, the efficacy of long context is inherently limited by the effective spans of the attention mechanisms in LLMs. Even with the Vicuna-13B-v1.5 model which supports a sequence length of 4k tokens, a clear reduction in improvements was found when adding much more information into the context. Therefore, future work needs to be done on improving the effectiveness of the context instead of only increasing the maximum allowed sequence length.

Due to resource constraints, experiments were only conducted on the SLURP corpus. Further studies will be carried out in the future on other speech-based slot-filling corpora using LLMs.

8 Ethics Statement

The approaches in this paper do not give rise to any additional risks beyond the ones directly inherited from the models. The ASR system might work more poorly for speakers from particular demographics and with particular accents. The framework also inherits the biases from all the language models used for experiments in this paper.

References

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

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proc. NeurIPS*.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's GPT-2 - how can I help you? towards the use of pre-trained language models for task-oriented dialogue systems. In *Proc. Workshop on Neural Generation and Translation*, pages 15–22.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *Proc. AAAI*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, et al. 2022. Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311.

Paul Francis Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Proc. NeurIPS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Xinya Du, Luheng He, Qi Li, Dian Yu, Panupong Pasupat, and Yuan Zhang. 2021. QA-driven zero-shot slot filling with weak supervision pretraining. In *Proc. ACL*, pages 654–664.

Gabor Fuisz, Ivan Vulić, Samuel Gibbons, Iñigo Casanueva, and Paweł Budzianowski. 2022. Improved and efficient conversational slot labeling through question answering. In *Proc. EMNLP*.

Michael Heck, Nurul Lubis, Benjamin Ruppik, Renato Vukovic, Shutong Feng, Christian Geishauser, Hsien-chin Lin, Carel van Niekerk, and Milica Gasic. 2023. ChatGPT for zero-shot dialogue state tracking: A solution or an opportunity? In *Proc. ACL*, pages 936–950.

Matthew Henderson and Ivan Vulić. 2021. ConVEx: Data-efficient and few-shot slot labeling. In *Proc. NAACL*, pages 3375–3389.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *Proc. ACL*, pages 1381–1393.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. In *Proc. ICLR*.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. In-context learning for few-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2627–2643.

Weizhe Lin, Bo-Hsiang Tseng, and Bill Byrne. 2021a. Knowledge-aware graph-enhanced gpt-2 for dialogue state tracking. In *Proc. EMNLP*.

Zhaojiang Lin, Bing Liu, Seungwhan Moon, Paul Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu, Andrea Madotto, Eunjoon Cho, and Rajen Subba. 2021b. Leveraging slot descriptions for zero-shot cross-domain dialogue state tracking. In *Proc. NAACL*.

Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. 2020. Coach: A coarse-to-fine approach for cross-domain slot filling. In *Proc. ACL*, pages 19–25.

Reham Omar, Omij Mangukiya, Panos Kalnis, and Essam Mansour. 2023. Chatgpt versus traditional question answering for knowledge graphs: Current status and future directions towards knowledge graph chatbots. *arXiv:2302.06466*.

OpenAI. 2023. Gpt-4 technical report. *arXiv:2303.08774*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, J. Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, P. Welinder, P. Christiano, J. Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. In *Proc. NeurIPS*.

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

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv:2302.12813*.

Alec Radford, Jong Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. In *Proc. NeurIPS*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog, 1(8), 9*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Anirudh Raju, Milind Rao, Gautam Tiwari, PRANAV DHERAM, Bryan Anderson, Zhe Zhang, Chul Lee, Bach Bui, and Ariya Rastrow. 2022. On joint training with interfaces for spoken language understanding. In *Proc. Interspeech*, pages 1253–1257.

Milind Rao, Pranav Dheram, Gautam Tiwari, Anirudh Raju, Jasha Droppo, Ariya Rastrow, and Andreas Stolcke. 2021. Do as i mean, not as i say: Sequence loss training for spoken language understanding. In *Proc. ICASSP*, pages 7473–7477.

Seunghyun Seo, Donghyun Kwak, and Bowon Lee. 2022. Integration of pre-trained networks with continuous token interface for end-to-end spoken language understanding. In *Proc. ICASSP*, pages 7152–7156.

Noam Shazeer. 2020. Glu variants improve transformer. *arXiv:2002.05202*.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *arXiv:2303.17580*.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2022. Roformer: Enhanced transformer with rotary position embedding. *arXiv:2104.09864*.

Guangzhi Sun, Chao Zhang, Ivan Vulić, Paweł Budzianowski, and Philip C. Woodland. 2023a. Knowledge-aware audio-grounded generative slot filling for limited annotated data. *arXiv:2307.01764*.

Guangzhi Sun, Chao Zhang, and Philip C. Woodland. 2023b. End-to-end spoken language understanding with tree-constrained pointer generator. In *Proc. ICASSP*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Pengwei Wang, Yinpei Su, Xiaohuan Zhou, Xin Ye, Liangchen Wei, Ming Liu, Yuan You, and Feijun Jiang. 2022. Speech2Slot: A Limited Generation Framework with Boundary Detection for Slot Filling from Speech. In *Proc. Interspeech*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *TMLR*.

Yangjun Wu, Han Wang, Dongxiang Zhang, Gang Chen, and Hao Zhang. 2022. Incorporating instructional prompts into a unified generative framework for joint multiple intent detection and slot filling. In *Proc. COLING*, pages 7203–7208.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proc. EMNLP*.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. In *Proc. NeurIPS*.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. 2023a. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv:2303.16199*.

Xiaoying Zhang, Baolin Peng, Kun Li, Jingyan Zhou, and Helen Meng. 2023b. Sgp-tod: Building task bots effortlessly via schema-guided llm prompting. *arXiv:2305.09067*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *lmsys blog*.

A Training set sizes

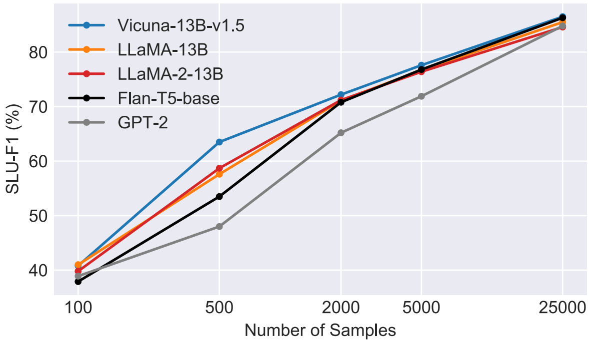

Figure 3: Variation of SLU-F1 on SLURP test set under reference ASR transcriptions against the number of samples in the training set for fine-tuning LLMs.

The influence of training set sizes on different LLMs and GPT-2 under Whisper medium ASR transcription is shown in Fig. 3.

B Influence of written form

Table 7: SLU-F1 comparison between written form (after Whisper text normalisation) and spoken form on standard SLURP test set using GPT-2 and Flan-T5-base model on reference transcriptions.

System	SLU-F1
GPT-2 Spoken	89.5
GPT-2	84.8
Flan-T5-base Spoken	90.3
Flan-T5-base	86.3

As shown in Table 7, written form in general had a lower SLU-F1 than spoken form. This was mainly because numbers that used to appear as several words were converted to a single word in written form (e.g. radio ninety-three point five ⇒ radio 93.5). As a result, the written form SLU-F1 was closer to the measurement of the F1 score of entire entities, and fewer partial credits were given to partly correct answers. This became more obvious when ASR transcriptions were used, causing a larger degradation than spoken form when using noisy transcriptions. Such degradation more honestly reflected the actual performance as a mistake in a word of a number, unlike other typo-like mistakes in names, would result in a different number.