SQATIN: Supervised Instruction Tuning Meets Question Answering for Improved Dialogue NLU

Anonymous ACL submission

Abstract

Task-oriented dialogue (ToD) systems help users execute well-defined tasks across a variety of domains (e.g., *flight booking* or *food ordering*), with their Natural Language Understanding (NLU) components being dedicated to the analysis of user utterances, predicting users' intents (*Intent Detection*, ID) and extracting values for informational slots (*Value Extraction*, VE). In most domains, labelled NLU data is scarce, making sample-efficient learning – enabled with effective transfer paradigms – paramount. In this work, we introduce SQATIN, a new framework for dialog NLU based on (i) instruction tuning and (ii) question-answering-based formulation of ID and VE tasks. According to the evaluation on established NLU benchmarks, SQATIN sets the new state of the art in dialogue NLU, substantially surpassing the performance of current models based on standard fine-tuning objectives in both in-domain training and cross-domain transfer, and it also surpasses off-the-shelf large language models for the same task, both in terms of performance and inference efficiency. Furthermore, SQATIN yields particularly large performance gains in cross-domain transfer, owing to the fact that our QA-based instruction tuning leverages similarities between natural language descriptions of classes (i.e., slots and intents) across domains.

1 Introduction

Task-oriented dialogue (ToD) systems support users in execution of specific, well-defined tasks through natural language interaction (e.g., ordering food or purchasing tickets) (Young, 2002; Budzianowski et al., 2018). Fine-grained understanding of user's utterances, commonly referred to as (dialogue) natural language understanding (NLU) is necessary for successful ToD (Larson et al., 2019; Casanueva et al., 2022). NLU modules of ToD systems typically solve two complementary tasks: (1) *Intent detection* (ID) aims to recognise the purpose (i.e., intent) of the user's utterance, classifying utterances into a set of predefined classes (e.g., the intent `lost_luggage` in *flight booking*); (2) *Value extraction* (VE) aims to extract spans that express values for any of the predefined informational slots (e.g., a dialog system for *booking flights* would have slots such as `origin`, `destination`, `time`, `maximal_price`). Realistic ToD setups for both ID and VE typically involve a relatively large number of labels (e.g., >100 different intent classes), commonly with a limited number of labelled instances per class. Successfully addressing these tasks thus amounts to enabling sample-efficient learning by means of transferring knowledge from other tasks (Gao et al., 2019), languages (Hung et al., 2022b; Moghe et al., 2023), or domains (Hung et al., 2022a; Moghe et al., 2023).

In recent years – in line with general NLP trends – most NLU models (Budzianowski and Vulić, 2019; Hosseini-Asl et al., 2020; Henderson and Vulić, 2021, inter alia) were obtained via standard, task-specific fine-tuning of pretrained Transformer-based language models (PLMs) (Devlin et al., 2019; Radford et al., 2019). Standard fine-tuning comes with task-specific (discriminative) objectives – different from LM-ing as the pretraining objective – which in principle impedes both knowledge transfer (1) from pretraining to downstream tasks and (2) between different downstream tasks. Prompting in contrast (Liu et al., 2023b) recasts downstream tasks into language modelling, making them more aligned with the models' pretraining. Finally, instruction-tuning (Sanh et al., 2022; Chung et al., 2022) – supervised training in which prompts created from instances are prepended with natural language descriptions of the tasks – facilitate the transfer between arbitrary tasks, leveraging the generalisation over task descriptions for zero-shot inference (i.e., inference for tasks unseen in training). Despite the impressive zero-shot and in-context few-shot inference abilities of the more recent Large

LMs (LLMs) (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023), supervised fine-tuning still brings substantial performance gains for dialog NLU (Hudeček and Dusek, 2023).

As generalisation to new domains (with limited in-domain annotation effort) is one of the main desiderata of ToD, some recent work on dialog NLU (Fuisz et al., 2022; Casanueva et al., 2022) has recognised that ID and VE can be cast as question answering (QA) tasks: this facilitates transfer from models trained on large QA datasets (Rajpurkar et al., 2016a; Lee et al., 2020), allowing also to capitalise on other large datasets previously recast as QA (McCann et al., 2018; Wang et al., 2022b). These efforts, however, amount to sequential transfer with standard fine-tuning for QA and thus (i) do not align their fine-tuning with the models' pretraining objective; and without an LM-based objective they (ii) cannot benefit from cross-task transfer via natural language task formulations.

Contributions. Motivated by the above observations, we propose a new framework for dialogue NLU driven by QA-based instruction tuning. In **SQATIN** (Supervised Question Answering Tuning on INstructions for dialogue NLU), we reformulate ID and VE into QA-based natural language instructions and, starting from a massively instruction-tuned PLM (Chung et al., 2022), fine-tune it for our tasks relying on a small number of in-domain examples. The rationale behind SQATIN is two-pronged: (1) transfer with a model that was previously instruction-tuned at scale improves the efficiency of learning from task-specific samples – this is highly desirable in most ToD domains, where one typically deals with only a handful of labelled utterances; (2) while small-scale ID/VE instruction-tuning specialises the model for a particular ToD domain (e.g., *restaurant booking*), the negligible size of in-domain training (compared to model's massive instruction-"pretraining") should prevent overfitting to the ToD training domain and allow for effective cross-domain transfer.

Our results strongly support both of the above assumptions: SQATIN yields state-of-the-art performance on two prominent dialogue NLU benchmarks both in in-domain and cross-domain evaluations. SQATIN brings particularly large gains in transfer between close ToD domains: classes in these domains have similar prompt descriptions, unlike the existing approaches based on standard fine-tuning. The code is available at `[URL]`.

2 SQATIN: Methodology

Standard Classification vs. Instruction Tuning for Dialog NLU. ID and VE are two tasks that comprise most Dialogue NLU modules. Both tasks are commonly cast as classification tasks: ID as a sequence classification task (i.e., one or more intent labels assigned for the whole utterance) and VE as a span extraction task, i.e., token-level classification.

In standard classification with pretrained LMs, a task-specific classifier $c_t : \mathbf{X} \in \mathbb{R}^h \mapsto \mathcal{P}(C_t)$ converts h-dimensional sequence or token representations (output by the LM) into a multinomial probability distribution over the set of task classes C_t. This means that a classifier c_t, trained for task t with classes C_t, cannot be used to make predictions for any other classification task t' with a different set of classes $C_{t'}$: thus, transfer between tasks can only occur indirectly through the parameters of the LM. This is particularly unfortunate for domain transfer in dialog NLU, where different domains often have semantically overlapping ID and VE classes (e.g., intent `confirm_order` is essentially the same intent in *flight booking* and in *food ordering*). In contrast, instruction-tuning recasts classification as a language modelling (i.e., generation) task $LM : \mathbf{x} \in \mathbb{R}^h \mapsto \mathcal{P}(V_t)$, with V_t as the subset of the LM's vocabulary where each token $v_t \in V_t$ represents one class c_t. This removes the need for a task-specific classifier (on top of the LM) and facilitates transfer between tasks, especially those with semantically overlapping class tokens.

QA-Based Instruction Tuning in SQATIN. For the above reasons, we adopt an instruction tuning approach to ID and VE. We start from models that have been instruction-tuned at scale (Wang et al., 2022a; Chung et al., 2022), since these models come with a strong inductive bias to complete any new task expressed as an instruction, exhibiting impressive generalisation abilities (i.e., good performance on new tasks).

As illustrated in Figure 1, we formulate both ID and VE as text-to-text tasks, with our instruction input consisting of (i) *context*, (ii) *instance*, and (ii) *prompt*. *Context* (e.g., *"The user says:"*) is the additional natural language description that is added (in our case, prepended) to the *instance*, a user's utterance; *Prompt* is the text that follows the *instance* and describes the actual task, that is, what is to be predicted from the instance. We formulate *prompts* as *questions* for both tasks. The motivation for this is the fact that the instruction-

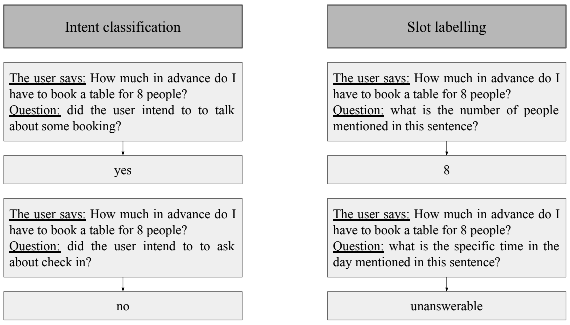

Figure 1: Instruction examples for ID and VE: for each we show one example where the class matches the utterance (i.e., for ID: correct intent class; for VE: a value for the slot class present) and one where it does not.

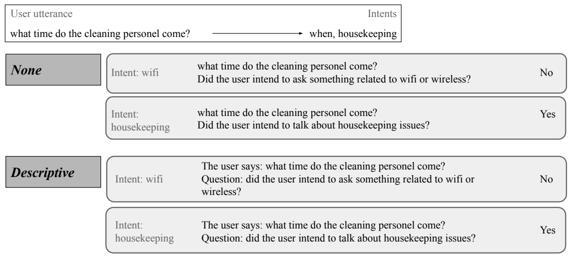

Figure 2: An annotated utterance from NLU++ transformed into corresponding SQATIN instruction instances. For brevity, we display the transformation for only two intents (wifi and housekeeping), but the same transformation was applied for all intents.

tuned model from which we start (Chung et al., 2022) has been pretrained on QA formulations of various tasks and thus comes with an inductive bias for answering questions. For each training utterance, we create one instruction-based training example for each of the intent and slot classes: (1) for ID, the question incorporates a natural language description of the intent class (e.g., *did the user intend to talk about some booking? corresponds to the intent class* booking) and requires a binary answer (yes or no); (2) for VE, the question incorporates a natural language description of an informational slot (e.g., *what is the number of people mentioned?* corresponds to the slot num_guests) – the expected answer is the value for that slot, as expressed in the instance or unanswerable if the instance does not contain a value for the slot.

A possible alternative to this "one instruction per instance and class" approach would be the more common prompt-based classification approach in which we create only one instruction per instance (e.g., with the question prompt *"what is the intent of this sentence?"*) and the model is expected to generate the token of the correct intent, choosing between tokens of all intent classes. This, however, comes with two major drawbacks: (i) ID tasks commonly come with a large number of classes (e.g., more than 50) – incorporating descriptions of all intent classes into a single prompt might thus surpass the input size of most models or they might struggle with memorizing all the options (Liu et al., 2023a); (ii) ID is, in principle, a multi-label, rather than multi-class problem, which means that utterances can express more than just one intent – this would require the model to output the text that somehow combines the tokens of more than one class, which is not something that instruction-based models have been pretrained for.

We experimented with two different instruction formulations: (1) without context (*None*), in which the instruction consists only of the instance and prompt; and (2) with descriptive context (*Desc.*, where we prepend the utterance with *"The user says:"* and the question prompt with *"Question:"*, as illustrated Figure 2. We selected these two particular instruction formulations (*None* and *Desc.*) based on their performance in a pilot study, which we describe in detail in the Appendix (A).

3 Experimental Setup

We rely on the Flan-T5 instruction-pretrained models (Chung et al., 2022). Unless stated otherwise, the main model is the Base variant. Training hyperparameters are described in detail in Appendix D.

Dialogue NLU Datasets. We run our experiments on two prominent dialogue NLU benchmarks: NLU++ (Casanueva et al., 2022) and CLINC-150 (Larson et al., 2019). NLU++ contains user utterances from real conversations in two domains: *banking* and *hotels*. NLU++ differs from most other ToD datasets in two important aspects: (i) it encompasses both generic (i.e., domain-universal) intents (e.g., booking) and slots (e.g., date) as well as the domain-specific ones (e.g., intent credit_card in the *banking* domain or slot no_rooms in the *hotels* domain) and (ii) its intents are "factorized" into "atomic" labels, with utterances then being assigned multiple intents (e.g., an utterance *"wanna change my room reservation"* is labelled with three atomic intents – *change*, *room*, and *booking* – rather than one complex intent change_room_booking). CLINC-150 encompasses over 20K utterances from 10 versatile domains (e.g., *travel*, *small talk*). Each domain has 15 intent labels, resulting in 150 intents in total. CLINC also contains utterances that do not belong to any of the 150 intents (labelled as `out_of_scope`). The fact that all CLINC domains have 15 intents, with the same number of instances per intent, allows for direct performance comparison across domains.[1] With few-shot fine-tuning in focus, we evaluate the models in a folded cross-validation setup. NLU++ already comes with predefined splits for 10-fold and 20-fold cross-validation.[2] Analogously, we split data from each CLINC domain in 10 folds, resulting in 150 training examples per fold.

Baselines. We compare SQATIN against two types of state-of-the-art models for dialogue NLU. For brevity, we provide training and model selection details for both baselines in the appendix.

Classification from Sentence Embeddings (CL-SE). Recent work on ID (Gerz et al., 2021; Casanueva et al., 2022) resorts to classifying – with a shallow feed-forward classifier – fixed sentence embeddings produced by of-the-shelf sentence encoders (SE). This avoids expensive fine-tuning of base LMs (e.g., RoBERTa) and yields comparable (or better) performance. We use LaBSE (Feng et al., 2022) as a state-of-the-art (SotA) SE.

Standard QA Fine-Tuning (QA-FT). Similar to us, these models adopt a QA-based formulation of dialogue NLU but exclude the instruction component (Namazifar et al., 2021; Casanueva et al., 2022; Fuisz et al., 2022). The key aspect is that the QA-based fine-tuning for ID and VE starts from the model that has previously been fine-tuned on large-scale QA datasets (e.g., SQUAD, Rajpurkar et al. (2016b, 2018)). To maximise comparability (given that SQATIN is based on Flan-T5), we obtain our QA-FT baseline by fine-tuning the T5 model (Raffel et al., 2020) previously trained on SQUAD 2.0.[3]

We report the standard micro-F1 scores. VE predictions are considered correct only if they exactly match the gold value span.

4 Main Evaluation

Preliminary Study: Zero-Shot ID & VE. The key hypothesis behind SQATIN is that instruction-tuned models have stronger inductive bias for dialogue NLU than models fine-tuned in the standard manner, including those trained for QA (Namazifar et al., 2021; Fuisz et al., 2022). We thus preliminarily compare zero-shot ID/VE performance of (1) the instruction-trained Flan-T5 and (2) T5 fine-tuned for QA on SQUAD2.0 (denoted QA-T5) on NLU++. The results in Table 1 show that Flan-T5 is much more robust "out of the box". While QA-T5 has better VE performance in the *banking* domain, it yields near-zero performance in all other setups. This validates our selection of the instruction-tuned Flan-T5 as the starting point for SQATIN.

Model	**ID**		**VE**	
	20-Fold	*10-Fold*	*20-Fold*	*10-Fold*
BANKING				
QA-T5	0.6	0.6	12.5	12.5
Flan-T5	21.9	21.9	3.2	3.2
HOTELS				
QA-T5	0.4	0.4	0.0	0.0
Flan-T5	20.9	21.9	5.9	5.8

Table 1: Zero-shot results for ID and VE on NLU++.

Model	*Templ.*	**ID**		**VE**	
		20-F	*10-F*	*20-F*	*10-F*
BANKING					
CL-SE		58.1	68.8	N/A	N/A
QA-FT: RoBERTa		80.3	85.6	50.5	56.7
QA-FT: mDeBERTa		80.8	85.0	59.7	66.5
QA-FT: T5		82.7	86.8	61.5	73.5
SQATIN	*None*	85.6	**88.5**	64.9	75.4
	Desc.	**85.8**	88.4	**66.3**	**76.3**
HOTELS					
CL-SE		51.9	61.8	N/A	N/A
QA-FT: RoBERTa		67.4	73.3	48.1	52.4
QA-FT: mDeBERTa		66.9	73.2	**61.6**	67.3
QA-FT: T5		69.2	76.5	57.2	**67.9**
SQATIN	*None*	73.1	78.0	58.0	**67.7**
	Desc.	**73.4**	**78.1**	58.7	67.0

Table 2: In-domain ID and VE performance for SQATIN and SotA baselines (CL-SE and QA-FT with different base models). **Bold:** best column score.

In-Domain Results. We next compare the supervised in-domain performance (i.e., training and test instances from the same domain) of SQATIN against the CL-SE and QA-FT baselines. Tables 2 and 3 display the results on NLU++ and CLINC-150, respectively. On NLU++, we additionally provide QA-FT results with two other base models, RoBERTa (Liu et al., 2019) and mDeBERTa (He et al., 2022), copied directly from (Casanueva et al., 2022) and (Moghe et al., 2023), respectively.

SQATIN consistently and considerably outperforms the baseline models, on both tasks and on both datasets. These results confirm that instruction-based models have stronger inductive biases than QA-fine-tuned models: these biases are propagated

[1] Prior work has mostly used CLINC-150 as a single-domain dataset with 150 intents, rather than multi-domain with domain-specific intents. In contrast, we are interested in cross-domain dialogue NLU performance and thus split the examples by domains. To ensure the replicability of results, we will make public the exact dataset splits that we used.

[2] In the 20-fold setup, one fold contains $\approx$ 100 utterances in the *banking* domain and $\approx$ 50 in the *hotels* domain.

[3] We use the checkpoint at `https://huggingface.co/mrm8488/t5-base-finetuned-squadv2`.

Model	Template	AUTO	BANKING	CREDIT CARD	HOME	KITCHEN &DINING	META	SMALL TALK	TRAVEL	UTILITY	WORK	**AVG**
CL-SE		92.74	92.30	90.48	88.58	91.19	90.19	90.90	95.29	94.53	91.93	91.81
QA-FT: T5		90.42	94.38	94.42	89.23	93.22	90.10	81.36	97.67	94.66	89.99	91.54
SQATIN	*None*	**94.47**	96.04	95.64	91.92	95.01	90.55	93.10	**97.77**	95.72	91.56	94.18
	Desc.	**94.47**	**96.11**	**95.85**	**92.66**	**95.36**	**91.52**	**93.12**	96.97	**96.07**	**92.01**	**94.42**

Table 3: In-domain ID results on CLINC-150 for SQATIN and the baselines (CL-SE and QA-FT).

Model	*Templ.*	**ID**		**VE**	
		20-F	*10-F*	*20-F*	*10-F*
BANKING → HOTELS					
QA-FT: T5		66.70	**69.68**	30.86	38.09
SQATIN	*None*	66.68	68.18	**33.24**	**39.48**
	Desc.	**67.04**	68.48	**33.24**	37.41
HOTELS → BANKING					
QA-FT: T5		59.76	66.12	35.08	44.60
SQATIN	*None*	65.35	67.34	44.72	**52.05**
	Desc.	**66.44**	**68.56**	**45.69**	51.87

Table 4: Domain transfer results for SQATIN and the QA-FT (T5) baseline on NLU++ (between BANKING and HOTELS). **Bold:** best score in each column.

in task-specific instruction-based fine-tuning, resulting in SotA performance. The gains seem more pronounced in setups with less training data (i.e., 20-Fold in Table 2) rendering instruction-tuning more sample efficient than (QA-based) fine-tuning. Overall, SQATIN seems to work slightly better with descriptive context prompts added to the instruction (compare *Desc.* vs. *None*).

Domain Transfer Results. We next train SQATIN in one (source) domain and apply it in another (target) domain. Table 4 and Figure 3 summarize the domain transfer results for NLU++ and CLINC-150 (all domain pairs), respectively.

Much like in in-domain training, SQATIN consistently outperforms the SoTA baseline QA-FT in domain transfer (the only exception is BANKING→HOTELS transfer for ID in the 10-Fold setup), only now by much wider margins for VE (e.g., by over 10 points in HOTELS→BANKING transfer in the 20-Fold setup). On CLINC-150, the results reveal not only that SQATIN consistently outperforms QA-FT (consistently lighter heatmap cells for SQATIN variants than for QA-T5) but that it is also able to better exploit label similarity between domains: e.g., for CREDIT CARD as the target domain, SQATIN obtains best performance when transferring from the BANKING domain, whereas QA-FT, in this case, finds AUTO as the best source.

Similarity of Intent Class Descriptions. Observing that SQATIN yields best transfer performance between intuitively related domains, we now investigate more closely what type of similarity between domains drives the transfer: (i) similarity of examples (sim-E) or (ii) similarity of intent class descriptions, incorporated in SQATIN's prompts (sim-C). We quantify sim-E as the average similarity across all pairs of utterances between the domains: with similarity of two utterances computed as cosine between their sentence embeddings, obtained with `mpnet` (Song et al., 2020) as the sentence encoder. Analogously, sim-C is computed as the average similarity of pairs of class prompts between the two domains. We then measure the correlation (Pearson's ρ) between the transfer performance and sim-E or sim-C. Table 5 shows these correlations for each CLINC-150 domain as transfer target. Correlations are largest for domains that do have related domains in the dataset (e.g., BANKING and CREDIT CARD) and lowest for domains that are quite different from all other (e.g., AUTO or UTILITY). Importantly, sim-C shows higher average correlation with transfer performance than sim-E: this suggests that SQATIN's instruction-based tuning with class descriptions in prompts truly captures similarities sets of intents and, consequently, especially improves transfer between related domains.

5 Further Analyses and Discussion

Cross-Task Generalisation. We next hypothesise that SQATIN facilitates transfer between the two dialogue NLU tasks, given that SQATIN's QA formulation conceptually allows for such cross-task transfer and presents both tasks to the model in the same format. Table 6 compares the zero-shot ID performance of the off-the-shelf Flan-T5 (*Non-tuned*) against the variant we SQATIN-fine-tune for VE. We observe substantial improvements in ID after instruction-tuning for VE (around 5% in the BANKING domain and over 10% in the HOTELS domain), proving effective cross-task generalisation of SQATIN in dialogue NLU.

We then fine-tune the models *jointly* on ID and VE. Table 7 compares single-task training vs. multi-task training on both tasks. While multi-task training yields no clear gains for ID (as the easier of the two tasks), it gives consistent gains for VE (0.5-1.5 F1 points). This again indicates that SQATIN facilitates transfer between the dialog NLU tasks.

Template	AUTO	BANKING	CREDIT CARD	HOME	KITCHEN &DINING	META	SMALL TALK	TRAVEL	UTILITY	WORK	AVG
In-Domain Training Examples											
None	-0.1443	0.5476	0.4268	0.1318	0.0204	0.0970	0.3279	0.0890	-0.2613	0.5451	0.2591
Desc.	-0.1069	0.5710	0.4695	-0.1121	0.1649	0.0929	0.1304	-0.3360	-0.35	0.6086	0.2942
Intent Descriptions											
None	-0.2600	0.6260	0.5076	0.3059	0.1208	0.2454	0.6019	0.1633	0.1388	0.3830	0.3353
Desc.	-0.3376	0.5533	0.5327	0.2319	-0.1091	0.3165	0.4884	0.1076	0.0449	0.4860	0.3208

Table 5: Correlation (Pearson's ρ) between domain transfer performance and domain similarity, measured in terms (i) of examples (sim-E) and (ii) class prompts (sim-C): shown for every CLINC-150 domain as the target.

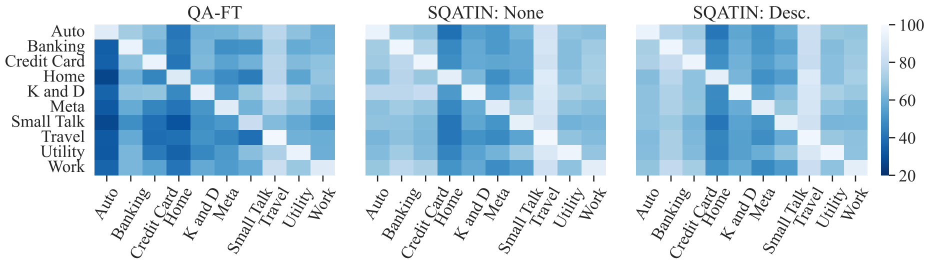

Figure 3: Cross-domain transfer results for ID on CLINC-150 for SQATIN and the SotA QA-FT baseline. Full results in the tabular format are in Appendix B. Diagonal values correspond to in-domain results. Source domains shown along the vertical axis and target domains along the horizontal axis.

Model	BANKING		HOTELS	
	20-Fold	*10-Fold*	*20-Fold*	*10-Fold*
Non-tuned	21.91	21.93	20.85	21.94
Tuned for VE	26.28	26.85	30.77	33.39

Table 6: SQATIN's (*Desc.* cross-task transfer performance on NLU++; VE→ID.

Model	*Template*		ID		VE	
			20-F	*10-F*	*20-F*	*10-F*
BANKING						
SQATIN	*None*	Single-task	85.55	88.53	64.92	75.41
		Multi-task	85.69	88.34	66.89	76.08
	Desc.	Single-task	85.78	88.41	66.32	76.26
		Multi-task	85.79	88.42	67.88	76.76
HOTELS						
SQATIN	*None*	Single-task	73.11	78.04	57.99	67.71
		Multi-task	72.70	77.73	61.27	68.66
	Desc.	Single-task	73.35	78.11	58.74	66.94
		Multi-task	73.15	77.74	61.74	68.66

Table 7: Cross-task transfer: comparison between (in-domain) single-task (ID *or* VE) and multi-task training (ID *and* VE) on NLU++.

Model Size. To analyse the impact of the underlying instruction-tuned model's size on performance, we also train SQATIN on top of the following Flan-T5 models: SMALL (80M parameters), BASE (250M) and LARGE (780M), with the scores provided in Appendix E. SQATIN yields strong in-domain performance even on top of the SMALL Flan-T5. The margin between LARGE and BASE is substantially smaller than that between BASE and SMALL; for in-domain ID, the gap between LARGE and BASE is negligible. The SMALL models performs notably worse than its larger siblings only in cross-domain transfer, especially for VE. Cross-domain performance of LARGE almost reaches the in-domain performance of SMALL, which is in line with observations that generalisation abilities of instruction-tuned models generally improve with their size (Chung et al., 2022).

Sample Efficiency. Due to large-scale instruction pretraining, we expect SQATIN to be more sample efficient than QA-FT and CL-SE. To test this, we train the models on training data of different sizes. The process is as follows: i) first, 1000 examples are randomly chosen for the test set; ii) from the rest we sample a random subset of N training examples; iii) models are then trained on training set from step ii) and evaluated on test set from step i). This ensures that models trained on sets of different sizes are evaluated on the same test set, making the performances comparable. We use the same hyperparameter configuration from §3 for all training sizes. Results in Figure 4 demonstrate that the scarcer the resources are, the more benefits SQATIN brings over the baselines (QA-FT and especially CL-SE). Another observation is that both QA-based approaches, QA-FT as well as SQATIN drastically outperform CL-SE in few-shot scenarios (cf. results for 32 and 64 training examples): this result justifies QA formulation for intent detection and value extraction in low-data setups.

Independent QA versus Multiple-Choice. By design SQATIN involves asking an independent

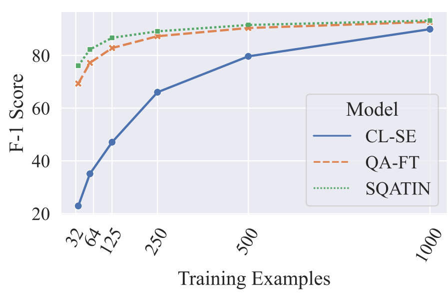

Figure 4: Comparison of ID models on BANKING domain on NLU++ for different training data sizes. The results are averages over 3 random seeds.

Model	Templ.	In-Domain		Cross-Domain	
		20-F	*10-F*	*20-F*	*10-F*
BANKING					
ChatGPT ZS	N/A	38.2	38.2	–	–
ChatGPT ICL	N/A	67.5	67.6	–	–
SQATIN	*None*	85.6	**88.5**	66.7	68.2
	Desc.	**85.8**	88.4	**67.0**	**68.5**
MC	*None*	62.0	67.9	39.3	46.1
	Desc.	63.9	68.5	42.5	47.7
HOTELS					
ChatGPT ZS	N/A	39.1	39.2	–	–
ChatGPT ICL	N/A	63.1	67.9	–	–
SQATIN	*None*	73.1	78.0	65.4	67.3
	Desc.	**73.4**	**78.1**	**66.4**	**68.6**
MC	*None*	45.5	58.2	37.3	50.8
	Desc.	50.0	59.7	41.3	51.9

Table 8: Standard SQATIN versus prompt-based multiple-choice (MC) task formulation for in-domain and cross-domain setups (ID on NLU++).

question about every intent (for ID) and every slot (VE) from the ontology for each user utterance: this decomposition might impact inference efficiency. A more efficient alternative might be a common multiple-choice prompt-based approach, where we create one instruction per utterance and provide the model with all possible intent classes or slots. The model is then expected to generate all intents or slot values that apply to the given utterance in a single response. We use the same instruction formulations to ensure comparability and represent possible intent classes with natural language descriptions (e.g., "to deny something", "to greet someone"); see an input example in Appendix F. Similarly to SQATIN, we finetune an instruction-tuned model, namely, Flan-T5 (BASE), on the MC-style input. Training hyperparameters are provided in Appendix D.

While offering potential benefits with inference speed, there are known deficiencies of this multiple-choice formulation (MC), as previously discussed in §2. For instance, the average length (in tokens) of input of the independent, binary SQATIN formulation for NLU++ ID and the MC formulation is 29.85 and 310.13, respectively. The difference might become even more salient with larger ontologies. The results for NLU++ in Table 8 demonstrate that the MC approach is considerably behind the independent-QA SQATIN both in in-domain and cross-domain setups, regardless of the training data size or template formulation. This indicates that the per-intent or per-slot independent question formulation is necessary for sample-efficient generalisation of SQATIN. We hypothesise that this is due to the data augmentation effects achieved this way.

SQATIN versus In-Context Learning with ChatGPT. One alternative to supervised tuning of smaller models is in-context learning (ICL) with much larger instruction-tuned language models. ICL could be more computationally efficient at training time as it does not require fine-tuning the model while being more demanding at inference time, as the model size is considerably larger. To compare the performance of ICL with SQATIN, we evaluate ChatGPT in two standard scenarios: (i) *zero-shot (ZS)*, when the provided instruction includes task description with all possible options (intent descriptions in our case); and (ii) *ICL*, when in addition to the above, the instruction also includes training examples which were used for supervised training in the models in every respective setting.[4] We evaluate `GPT-3.5-turbo-instruct` as the underlying model due to its strong ICL capabilities (Ye et al., 2023).

Results in Table 8 demonstrate that SQATIN performs consistently better than ChatGPT in both ZS and ICL scenarios. This suggests that even large models with ICL (and higher inference demands and cost) cannot surpass smaller highly specialised SQATIN models for the fine-grained dialogue NLU tasks such as the NLU++ ones.

Parameter Efficiency. Next, we also investigate whether the performance benefits of SQATIN extend when we replace full-model fine-tuning with the standard parameter-efficient fine-tuning (PEFT) methods (Ruder et al., 2022) such as *adapters* (Houlsby et al., 2019; Pfeiffer et al., 2021). In our case, relying on the standard bottleneck adapters with the reduction factor of 16 (Poth et al., 2023), for Flan-T5 BASE, the number of tunable parameters is $\approx 250\times$ smaller than the size of the original model. The hyperparameters and training procedure are the same (see §3), except for the

[4]For the *10-Fold* setup including all examples was impossible due to the context length limit. In this case, we fitted as many examples as possible by the context length.

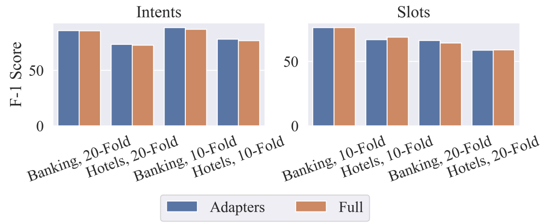

Figure 5: Full-model fine-tuning (≈ 248M tunable parameters) versus PEFT with Adapters (≈ 1.8M tunable parameters) in in-domain ID and VE.

learning rate which was increased to $5e$-4.[5] Figure 5 displays the performance of adapter-based fine-tuning on NLU++. The results render adapters extremely effective, yielding results comparable to those of full fine-tuning, indicating that the benefits of SQATIN are not limited to full-model fine-tuning only.

6 Related Work

Pretraining for ToD Dialogue. LLMs, trained on large web-scale corpora, revolutionised NLP, bringing massive performance gains to most NLP tasks. Besides general corpora, the most successful pretrained LMs for dialogue have have been additionally trained on more specialised, conversation-like data (e.g., from Reddit or Twitter). These models have been increasingly successful in both open-domain (Adiwardana et al., 2020; Bao et al., 2021; Thoppilan et al., 2022; Dettmers et al., 2023, inter alia) and task-oriented dialogue (Budzianowski and Vulić, 2019; Lin et al., 2020; Ham et al., 2020; Zhao et al., 2020). Compared to general-purpose LM pretraining (e.g., BERT), dialogic pretraining has been shown to lead to higher performance in cross-domain transfer for dialogue NLU tasks (Mi et al., 2021; Lin et al., 2021; Hung et al., 2022a, interalia) due to the versatility of texts used in pretraining. Another stand of work incestigated multi-task learning setups for dialogue NLU (Hosseini-Asl et al., 2020; Liu et al., 2021; Su et al., 2022). In this work, in contrast, we resorted to models *pre-trained* on multiple tasks with instruction-based objectives, resulting with stronger inductive biases for cross-domain and cross-task settings. To the best of our knowledge, this work is the first to propose a unified (QA- and instruction-based) framework for both dialogue NLU tasks (ID and VE).

Instruction Tuning for Dialogue NLU. Instruction tuning is an emergent framework in NLP where a generative model completes a task by following natural language *instructions*, possibly including few labelled instances following the instruction to make the whole prompt. These models generalise particularly well to tasks unseen during training (Chung et al., 2022; Chowdhery et al., 2023) due to their ability to leverage the information about a task during inference (Liu et al., 2023b). The performance, especially in zero-shot setup, is highly dependent on task definitions (Liu et al., 2023b) or providing several training examples (Min et al., 2022) in the instruction text (commonly known as in-context learning). Dialogue follows the same trend: recent work (Gupta et al., 2022) demonstrated the zero-shot effectiveness of instruction-tuned models on dialogue tasks. Instruction engineering (Gupta et al., 2022; Ruder et al., 2023) and increasing the number of in-context instances can further improve the models' performance (Madotto et al., 2021; Mi et al., 2022). The input (context) size of the models, however, puts a limit on the number of (1) training examples (2) classes (i.e., their descriptions) one can include in the prompt. SQATIN deals with the issue in two ways: a) by recasting the dialogue NLU tasks as independent QA, at inference time we remove the need for the model to see all class descriptions at once; and b) we allow the model to learn from training examples in supervised fashion (versus in-context) thus not being limited by the base model's input length. We empirically validate that both have strong positive impact on task performance.

7 Conclusion

We have introduced a novel framework for dialogue NLU, SQATIN, which combined (i) supervised instruction tuning and (ii) question-answering formulation of intent detection and value extraction. We evaluated SQATIN on two established dialogue NLU benchmarks, demonstrating that SQATIN brings substantial and consistent improvements over the existing SoTA approaches. The performance gains are especially pronounced in cross-domain transfer, as SQATIN can leverage similarities between classes across domains via their descriptions. SQATIN also performs well in cross-task transfer, enabling the two dialogue NLU tasks to benefit from one another. We also show that SQATIN supports parameter-efficient fine-tuning and that it largely outperforms ICL with much larger (and more expensive) language models.

[5] Grid search over the set {$5e$-5, $5e$-4, $5e$-3} was run.

Limitations

Our experiments are based on the Flan collection of models as they were pretrained on a wide collection of tasks. However, we note that there are other instruction-based models (Ouyang et al., 2022; Sanh et al., 2022; Zhang et al., 2022, inter alia), with more getting published almost on a daily basis, which could be used with the proposed method and the choice of the instruction-based model is orthogonal to the proposed methodology. We leave wider exploration in this direction as future work.

Additionally, we have focused on a single-source transfer across domains, i.e., a model trained on one domain was expected to be able to transfer to a multitude of others. Future work will also explore the multi-source cross-domain transfer where the model would be finetuned on combined data from several domains and tested on data from domains not included in training.

In the evaluation, we rely on available standard dialogue NLU benchmarks built specifically to test few-shot in-domain and cross-domain generalisation abilities of the models. It is important to note that the benchmarks are only for English dialogue NLU. We opt to confirm the effectiveness of SQATIN in multilingual settings in future work. Exploration of SQATIN in multilingual settings would be also dependent on the availability of strong multilingually pretrained instruction-based models.

Lastly, due to the computational cost of finetuning instruction-based models we largely rely on instruction wordings and training hyperparameters from prior work. We hope to perform a more detailed hyperparameter search in both wording of the instructions and training hyperparameters in the future.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *ArXiv preprint*, abs/2001.09977.

Siqi Bao, Huang He, Fan Wang, Hua Wu, Haifeng Wang, Wenquan Wu, Zhen Guo, Zhibin Liu, and Xinchao Xu. 2021. PLATO-2: Towards building an open-domain chatbot via curriculum learning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2513–2525, Online. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's GPT-2 - how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22, Hong Kong. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Inigo Casanueva, Ivan Vulić, Georgios Spithourakis, and Paweł Budzianowski. 2022. NLU++: A multi-label, slot-rich, generalisable dataset for natural language understanding in task-oriented dialogue. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1998–2013, Seattle, United States. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *ArXiv preprint*, abs/2210.11416.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems 2023 (to appear)*.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Gabor Fuisz, Ivan Vulić, Samuel Gibbons, Inigo Casanueva, and Paweł Budzianowski. 2022. Improved and efficient conversational slot labeling through question answering. *ArXiv preprint*, abs/2204.02123.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 264–273, Stockholm, Sweden. Association for Computational Linguistics.

Daniela Gerz, Pei-Hao Su, Razvan Kusztos, Avishek Mondal, Michał Lis, Eshan Singhal, Nikola Mrkšić, Tsung-Hsien Wen, and Ivan Vulić. 2021. Multilingual and cross-lingual intent detection from spoken data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7468–7475, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey Bigham. 2022. InstructDial: Improving zero and few-shot generalization in dialogue through instruction tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 505–525. Association for Computational Linguistics.

Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592, Online. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2022. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Matthew Henderson and Ivan Vulić. 2021. ConVEx: Data-efficient and few-shot slot labeling. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3375–3389, Online. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Vojtěch Hudeček and Ondrej Dusek. 2023. Are large language models all you need for task-oriented dialogue? In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 216–228, Prague, Czechia. Association for Computational Linguistics.

Chia-Chien Hung, Anne Lauscher, Simone Ponzetto, and Goran Glavaš. 2022a. DS-TOD: Efficient domain specialization for task-oriented dialog. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 891–904, Dublin, Ireland. Association for Computational Linguistics.

Chia-Chien Hung, Anne Lauscher, Ivan Vulić, Simone Ponzetto, and Goran Glavaš. 2022b. Multi2WOZ: A robust multilingual dataset and conversational pretraining for task-oriented dialog. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3687–3703, Seattle, United States. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Gyeongbok Lee, Seung-won Hwang, and Hyunsouk Cho. 2020. SQuAD2-CR: Semi-supervised annotation for cause and rationales for unanswerability in SQuAD 2.0. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5425–5432, Marseille, France. European Language Resources Association.

Zhaojiang Lin, Bing Liu, Andrea Madotto, Seungwhan Moon, Zhenpeng Zhou, Paul Crook, Zhiguang Wang, Zhou Yu, Eunjoon Cho, Rajen Subba, and Pascale Fung. 2021. Zero-shot dialogue state tracking via cross-task transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7890–7900, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. MinTL: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405, Online. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts. *ArXiv preprint*, abs/2307.03172.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Qi Liu, Lei Yu, Laura Rimell, and Phil Blunsom. 2021. Pretraining the noisy channel model for task-oriented dialogue. *Transactions of the Association for Computational Linguistics*, 9:657–674.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Andrea Madotto, Zhaojiang Lin, Genta Indra Winata, and Pascale Fung. 2021. Few-shot bot: Prompt-based learning for dialogue systems. *ArXiv preprint*, abs/2110.08118.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *ArXiv preprint*, abs/1806.08730.

Fei Mi, Yasheng Wang, and Yitong Li. 2022. CINS: comprehensive instruction for few-shot learning in task-oriented dialog systems. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11076–11084. AAAI Press.

Fei Mi, Wanhao Zhou, Lingjing Kong, Fengyu Cai, Minlie Huang, and Boi Faltings. 2021. Self-training improves pre-training for few-shot learning in task-oriented dialog systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1887–1898, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

Nikita Moghe, Evgeniia Razumovskaia, Liane Guillou, Ivan Vulić, Anna Korhonen, and Alexandra Birch. 2023. Multi3NLU++: A multilingual, multi-intent, multi-domain dataset for natural language understanding in task-oriented dialogue. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3732–3755.

Mahdi Namazifar, Alexandros Papangelis, Gokhan Tur, and Dilek Hakkani-Tür. 2021. Language model is all you need: Natural language understanding as question answering. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7803–7807. IEEE.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Clifton Poth, Hannah Sterz, Indraneil Paul, Sukannya Purkayastha, Leon Engländer, Timo Imhof, Ivan Vulić, Sebastian Ruder, Iryna Gurevych, and Jonas Pfeiffer. 2023. Adapters: A unified library for parameter-efficient and modular transfer learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 149–160, Singapore. Association for Computational Linguistics.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016a. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016b. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Sebastian Ruder, Jonathan H Clark, Alexander Gutkin, Mihir Kale, Min Ma, Massimo Nicosia, Shruti Rijhwani, Parker Riley, Jean-Michel A Sarr, Xinyi Wang, et al. 2023. Xtreme-up: A user-centric scarce-data benchmark for under-represented languages. *ArXiv preprint*, abs/2305.11938.

Sebastian Ruder, Jonas Pfeiffer, and Ivan Vulić. 2022. Modular and parameter-efficient fine-tuning for NLP models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 23–29, Abu Dubai, UAE. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. Multi-task pre-training for plug-and-play task-oriented dialogue system. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4661–4676, Dublin, Ireland. Association for Computational Linguistics.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *ArXiv preprint*, abs/2201.08239.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022a. Benchmarking generalization via in-context instructions on 1,600+ language tasks. *ArXiv preprint*, abs/2204.07705.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022b. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *ArXiv preprint*, abs/2303.10420.

Steve J Young. 2002. Talking to machines (statistically speaking). In *INTERSPEECH*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *ArXiv preprint*, abs/2205.01068.

Yufan Zhao, Can Xu, and Wei Wu. 2020. Learning a simple and effective model for multi-turn response generation with auxiliary tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3472–3483, Online. Association for Computational Linguistics.

A Different Instruction Formulations

Choosing the right instruction formulation is often crucial (or at least important) to obtain strong performance from the instruction-based models. Thus, we conducted a pilot study for picking an optimal one. We experiment with 4 *context* options, 4 options of text preceding a question and 3 *prompt* options. The options (shown in Table 9) were adapted from the templates used to train the Flan models (Chung et al., 2022). We use Fold-0 of 10-Fold in-domain setting for intent detection to determine the best instruction formulation.

The results of the preliminary study are shown in Table 10. Although the range of results is not that large, we focus on two instruction formulations in further experiments: `none-none-none` and `usersaid-question-none`. The former is picked for similarity with the simple question answering formulation, although it leads to a lower performance. This enables direct comparison to QA-based models. As this formulation contains only the input sentence and the questions (no description of the task or its context), we denote it as *None*. The former instruction formulation (`usersaid-question-none`) is used as it contains the description of the context of the task and it led to the highest performance in the pilot study. As it contains a short description of the task, we denote it as *Descriptive (Desc.)*.

B Full Cross-Domain Results on CLINC-150 for Different Base Models

The cross-domain results on CLINC-150 for QA-FT and different versions of SQATIN are provided in Tables 11, 12 and 13.

C Comparison of Single-Task and Multi-Task Models for Cross-Domain Setups

Comparison of cross-domain results of models trained with SQATIN in single-task and multi-task setings is shown in Table 14.

Context
• “” `[none]` • “Given the following sentence: ” `[given]` • “Sentence: ” `[sent]` • “The user says: ” `[usersaid]`
Pre-question
• “” `[none]` • “Question: ” `[question]` • “Based on the question: ” `[based]` • “Based on the question above: ” `[basedabove]`
Prompt
• “” `[none]` • “Answer: ” `[answer]` • “Options: -yes -no Answer:” `[answeroptions]`

Table 9: Variants of instruction formulation.

D Fine-tuning and Hyperparameters

The classifier of the CL-SE baseline is a feed-forward network with a single hidden layer of dimensionality 512 and *tanh* as the non-linear activation function. With multi-label formulations of classification tasks (because instances in NLU++ can have multiple labels and those in CLINC-150 none), we apply *sigmoid* as an output activation and train with the binary cross-entropy loss. At inference, we consider an intent class to be predicted if its probability, output of the sigmoid activation, is above the threshold $\theta = 0.3$.

The models are implemented using Transformers library (Wolf et al., 2020). The models are loaded with sequence-to-sequence language modeling head. Baseline QA-based models and SQATIN are fine-tuned with the same protocol and hyperparameters as in prior work (Casanueva et al., 2022; Fuisz et al., 2022; Moghe et al., 2023). They are trained for 10 epochs with the batch size of 8, with Adam optimizer (Kingma and Ba, 2015) and the learning rate of 5e-5. Unless stated differently, we report the average cross-validation performance across all 10 or 20 folds the results are averages of 10 and 20 runs for 10- and 20-Fold setups, respectively.[6]

[6]We focus on the pre-defined few-shot 10-Fold and 20-Fold setups, as the baselines already demonstrate saturated performance on Large training data setups (Casanueva et al., 2022).

Context	Pre-question	Prompt	Banking	Hotels	AVG
none	none	none	77.2	67.3	72.25
sent	none	none	81.31	76.45	78.88
none	none	answer	80.96	77.14	79.05
given	none	none	81.4	76.96	79.18
none	none	answer-options	81.22	77.26	79.24
none	based-above	answer	82.65	75.9	79.28
usersaid	none	none	81.72	77.35	79.54
given	none	answer	81.49	77.69	79.59
sent	none	answer	81.36	78.06	79.71
none	based	answer	82.1	77.33	79.72
none	based	answer-options	82.1	77.37	79.74
sent	based	none	82.13	77.38	79.76
sent	based-above	none	82.68	77	79.84
sent	based-above	answer	82.73	77.06	79.90
sent	based	answer	82.15	77.74	79.95
none	based-above	answer-options	82.67	77.24	79.96
sent	none	answer-options	81.4	78.63	80.02
none	based	none	82.08	78.1	80.09
usersaid	based	none	82.34	77.92	80.13
usersaid	none	answer-options	82.05	78.28	80.17
given	none	answer-options	81.7	78.63	80.17
given	question	answer	83.49	76.94	80.22
sent	based-above	answer-options	82.8	77.65	80.23
none	based-above	none	82.57	77.93	80.25
none	question	answer	83.17	77.35	80.26
sent	question	none	83.25	77.27	80.26
usersaid	based	answer	82.39	78.15	80.27
sent	question	answer	83.39	77.29	80.34
usersaid	based	none	82.99	77.72	80.36
usersaid	based	answer	83.05	77.68	80.37
none	question	answer-options	83.22	77.61	80.42
given	question	answer-options	83.6	77.39	80.50
usersaid	none	answer	81.83	79.17	80.5
sent	based	answer-options	82.29	78.78	80.56
given	question	none	83.42	77.66	80.54
usersaid	based	answer-options	82.42	78.67	80.55
sent	question	answer-options	83.4	77.7	80.55
usersaid	based	answer-options	83.08	78.44	80.76
none	question	none	83.08	78.5	80.79
usersaid	question	answer	83.88	77.74	80.81
usersaid	question	answer-options	84.2	77.43	80.82
usersaid	question	none	83.85	78.07	80.96

Table 10: Performance of SQATIN with different instruction wordings. The options are ordered in ascending average order.

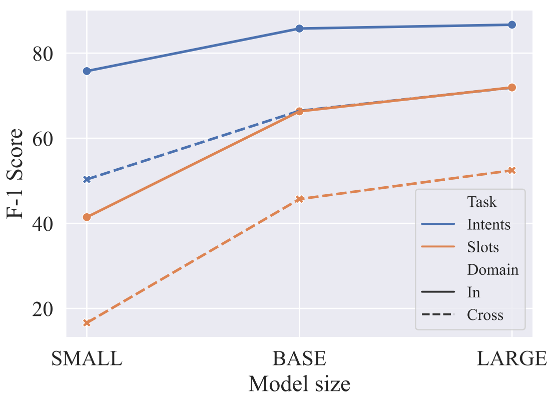

Figure 6: ID and VE performance (BANKING domain of NLU++, 20-Fold setup) for SQATIN trained on top of Flan-T5 models of different sizes. Similar trends are observed in the HOTELS domain, see Figure 7.

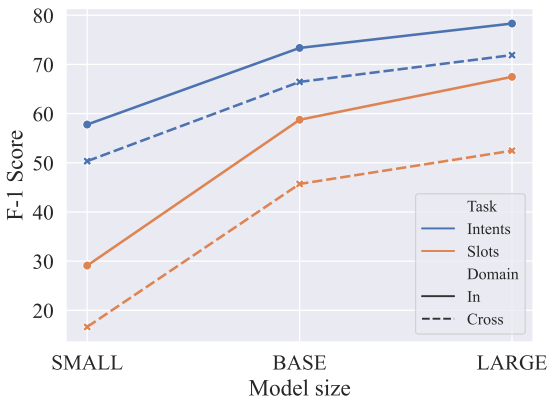

Figure 7: ID and VE performance (HOTELS domain of NLU++, 20-Fold setup) for SQATIN trained on top of Flan-T5 models of different sizes.

```
The user says: we will arrive tomorrow at 25 to 7
p.m.

Question: what did the user intend to ask?
Include all applicable options. Split the outputs
with $$.

Options:
to affirm something
to deny something
to say I don't know
to acknowledge what was said
to greet someone
<...>
to ask something related to wifi or wireless
to ask something related to gym
to ask something related to spa or beauty services
to ask something related to some room amenities
to talk about housekeeping issues
to talk about room service

Answer:
```

Figure 8: Input example for the multiple-choice formulation in the ID task.

E Results for Different Model Sizes

The results for different model sizes for the two domains of NLU++ are plotted in Figure 6 and Figure 7.

F Instructions with the Multiple Choice Formulation

Figure 8 shows an example of the multiple choice formulation for the ID task, including the instruction text, user query example and all possible options for the answers.

	QA-FT pretrained on SQUAD 2.0									
	AUTO	BANKING	CREDIT CARD	HOME	K AND D	META	SMALL TALK	TRAVEL	UTILITY	WORK
AUTO	90.42	71.08	65.22	42.03	61.23	61.78	65.64	77.04	66.7	60.5
BANKING	34.67	94.38	62.16	43.35	62.51	49.43	50.35	74.33	58.96	61.45
CREDIT CARD	35.19	66.94	94.42	41.28	64.05	55.86	61.13	76.54	64.14	66.92
HOME	26.68	60.4	46.07	89.23	55.95	48.64	43.35	76.05	56.65	68.08
K AND D	35.96	66.85	67.75	46.98	93.22	54.52	68.6	80.95	71.08	65.5
META	32.51	58.92	45.94	41.11	51.25	90.1	61.68	74.11	67.33	58.19
SMALL TALK	27.2	49.17	39.61	30.69	49.17	52.4	81.36	64.59	58.16	51.62
TRAVEL	32.96	58.54	38.89	39.71	50.6	46.53	39.46	97.67	61.13	59.72
UTILITY	32.61	63.12	42.76	35.91	46.87	52.67	65.77	73.62	94.65	60.08
WORK	36.32	62.9	55.93	41.05	58.24	53.14	58.62	81.83	69.13	89.99

Table 11: *Cross-domain* intent detection using QA-based model on CLINC-150 (Larson et al., 2019). K AND D stands for KITCHEN AND DINING domain. The rows are source domains while columns show target domains.

	SQATIN: *None*									
	AUTO	BANKING	CREDIT CARD	HOME	K AND D	META	SMALL TALK	TRAVEL	UTILITY	WORK
AUTO	94.47	70.87	67.26	39.75	54.96	52.2	61.57	85.01	67.09	65.71
BANKING	71.2	96.04	74.53	46.92	58.31	52.81	58.3	86.02	65.58	70.27
CREDIT CARD	70.08	77.44	95.64	48.97	58.71	57	58.4	84.3	65.53	71.68
HOME	65.8	76.24	68.91	91.91	63.3	49.18	56.1	89.59	66.98	72.51
K AND D	77.25	77.38	79.84	52.53	95.01	56.22	67.09	88.01	72.75	69.7
META	66.5	70.49	67.33	46.85	59.05	90.55	71.51	85.98	67.26	65.47
SMALL TALK	67.36	67.07	63.8	41.52	57.04	51.12	93.1	83.94	61.43	62.68
TRAVEL	62.8	66.26	63.34	41.94	50.58	47.71	55.97	97.77	67.35	64.58
UTILITY	64.6	70.71	64.35	45.68	55.88	61.6	70.91	88.28	95.72	67.97
WORK	68.68	77.19	73.12	50.89	58.03	48.63	54.5	83.31	67.05	91.56

Table 12: *Cross-domain* intent detection using SQATIN on CLINC-150 (Larson et al., 2019) with *None* templates. K AND D stands for KITCHEN AND DINING domain. The rows are source domains while columns show target domains.

	SQATIN: *Desc.*									
	AUTO	BANKING	CREDIT CARD	HOME	K AND D	META	SMALL TALK	TRAVEL	UTILITY	WORK
auto	94.47	75.69	70.47	41.68	56.88	50.47	59.61	82.45	68.54	67.51
banking	72.43	96.11	75.91	46.77	59.13	51.44	55.68	81.96	65.14	69.08
credit card	73.62	80.39	95.85	49.55	61.13	54.34	60.59	80.81	66.01	70.23
home	65.04	76.7	66.99	92.66	62.81	49.83	54.21	88.98	66.03	72.07
k and d	66.79	73.88	66.92	47.91	95.36	57.31	65.57	87.18	72.71	69.37
meta	66.73	73.66	67.55	47.56	59.12	91.52	68.59	86.31	67.01	63.85
small talk	67.08	69.89	61.95	41.26	55.93	51.33	93.12	84.28	62.62	62.97
travel	64.5	73.05	63.56	46	54.73	48.81	59.14	96.97	68.92	66.66
utility	65.39	73.03	64.25	45.66	55.26	59.82	68.29	87.59	96.07	67.09
work	67.8	79.15	71.26	50.41	58.86	47.48	53.41	82.07	67.15	92.01

Table 13: *Cross-domain* intent detection using SQATIN on CLINC-150 (Larson et al., 2019) with *Descriptive* templates. K AND D stands for KITCHEN AND DINING domain. The rows are source domains while columns show target domains.

Model	Template		ID		VE	
			20-Fold	10-Fold	20-Fold	10-Fold
BANKING → HOTELS						
SQATIN	*None*	Single-Task	66.61	68.18	33.24	39.48
		Multi-Task	66.73	68.59	33.81	39.77
	Desc.	Single-Task	67.04	68.48	33.25	37.41
		Multi-Task	67.28	68.15	33.08	36.18
HOTELS → BANKING						
SQATIN	*None*	Single-Task	65.35	67.34	44.72	52.05
		Multi-Task	64.68	67.06	45.38	51.44
	Desc.	Single-Task	66.44	68.56	45.69	51.87
		Multi-Task	66.86	68.08	46.02	52.04

Table 14: Comparison of single-task and multi-task models for cross-domain intent detection and value extraction on NLU++.