X2Parser: Cross-Lingual and Cross-Domain Framework for Task-Oriented Compositional Semantic Parsing

Zihan Liu, Genta Indra Winata, Peng Xu, Pascale Fung
Center for Artificial Intelligence Research (CAiRE)
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
zihan.liu@connect.ust.hk, pascale@ece.ust.hk

Abstract

Task-oriented compositional semantic parsing (TCSP) handles complex nested user queries and serves as an essential component of virtual assistants. Current TCSP models rely on numerous training data to achieve decent performance but fail to generalize to low-resource target languages or domains. In this paper, we present **X2Parser**, a transferable **Cross**-lingual and **Cross**-domain **Parser** for TCSP. Unlike previous models that learn to generate the hierarchical representations for nested intents and slots, we propose to predict flattened intents and slots representations separately and cast both prediction tasks into sequence labeling problems. After that, we further propose a fertility-based slot predictor that first learns to dynamically detect the number of labels for each token, and then predicts the slot types. Experimental results illustrate that our model can significantly outperform existing strong baselines in cross-lingual and cross-domain settings, and our model can also achieve a good generalization ability on target languages of target domains. Furthermore, our model tackles the problem in an efficient non-autoregressive way that reduces the latency by up to 66% compared to the generative model.[1]

1 Introduction

Virtual assistants can perform a wide variety of tasks for users, such as setting reminders, searching for events, and sending messages. Task-oriented compositional semantic parsing (TCSP) which comprehends users' intents and detects the key information (slots) in the utterance is one of the core components in virtual assistants. Existing TCSP models highly rely on large amounts of training data that usually only exist in high-resource domains and languages (e.g., English), and they generally fail to generalize well in a low-resource scenario. Given that collecting enormous training data is expensive and time-consuming, we aim to develop a transferable model that can quickly adapt to low-resource target languages and domains.

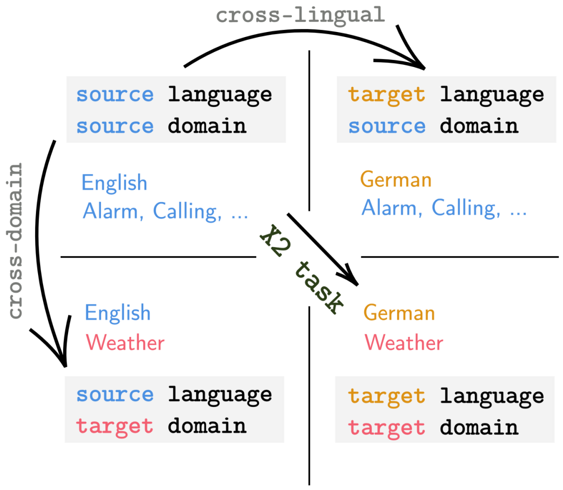

Figure 1: Illustration of the cross-lingual task, cross-domain task, and the combination of both (X2 task).

The traditional semantic parsing can be treated as a simple joint intent detection and slot filling task (Liu and Lane, 2016; Goo et al., 2018; Zhang et al., 2019), while compositional semantic parsing has to cope with complex nested queries, which requires more sophisticated models. Current state-of-the-art TCSP models (Rongali et al., 2020; Li et al., 2020a) are generation-based models that learn to directly generate the hierarchical representations which contain nested intent and slot labels.[2] We argue that the hierarchical representations are relatively complex, and the models need to learn when to generate the starting intent or slot label, when to copy tokens from the input, and when to generate the end of the label. Hence, large quantities of train-

[1] The code will be released in https://github.com/zliucr/X2Parser.

[2] An example of hierarchical representations is illustrated at the bottom of Figure 2.

Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021), pages 112–127
Bangkok, Thailand (Online), August 6, 2021. ©2021 Association for Computational Linguistics

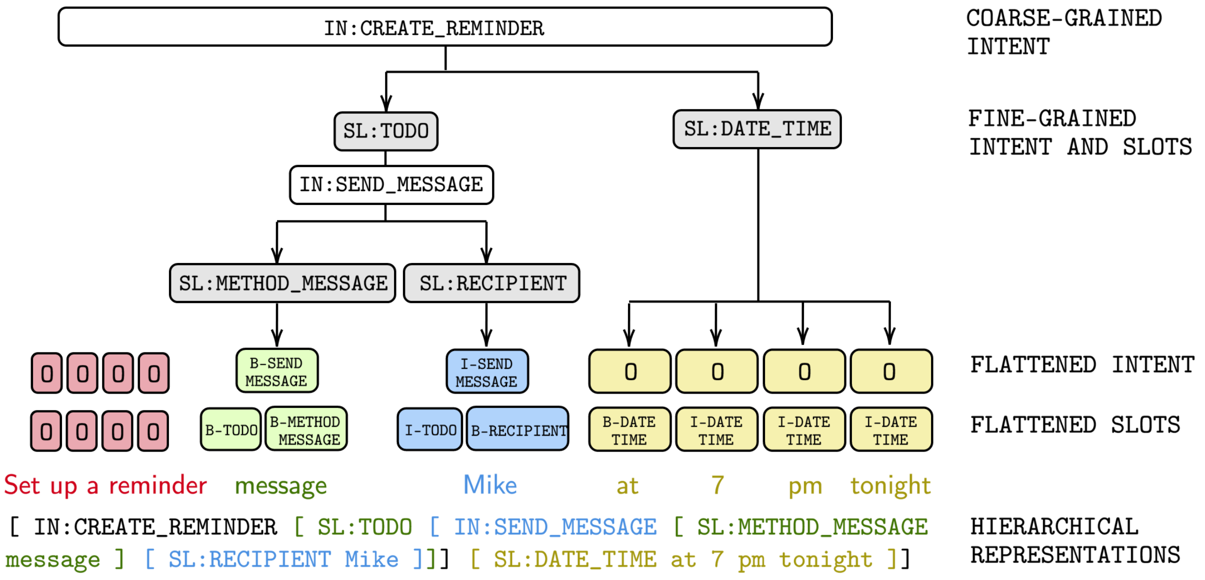

Figure 2: One data example with the illustration of our proposed flattened intents and slots representations, as well as the hierarchical representations used in Li et al. (2020a).

ing data are necessary for the models to learn these complicated skills (Rongali et al., 2020), while they cannot generalize well when large datasets are absent (Li et al., 2020a). Moreover, the inference speed of generation-based models will be greatly limited by the output length.

In this paper, we propose a transferable cross-lingual and cross-domain parser (X2Parser) for TCSP. Instead of generating hierarchical representations, we convert the nested annotations into flattened intent and slot representations (as shown in Figure 2) so that the model can learn to predict the intents and slots separately. We cast the nested slot prediction problem into a special sequence labeling task where each token can have multiple slot labels. To tackle this task, our model first learns to predict the number of slot labels, which helps it capture the hierarchical slot information in user queries. Then, it copies the corresponding hidden state for each token and uses those hidden states to predict the slot labels. For the nested intent prediction, we cast the problem into a normal sequence labeling problem where each token only has one intent label since the nested cases for intents are simpler than those for slots. Compared to generation-based models (Li et al., 2020a), X2Parser simplifies the problem by flattening the hierarchical representations and tackles the task in a non-autoregressive way, which strengthen its adaptation ability in low-resource scenarios and greatly reduce the latency.

As shown in Figure 1, we conduct experiments on three low-resource settings: cross-lingual, cross-domain, and a combination of both. Results show that our model can remarkably surpass existing strong baselines in all the low-resource scenarios by more than 10% exact match accuracy, and can reduce the latency by up to 66% compared to generation-based models. We summarize the main contributions of this paper as follows:

- We provide a new perspective to tackle the TCSP task, which is to flatten the hierarchical representations and cast the problem into several sequence labeling tasks.
- X2Parser can significantly outperform existing strong baselines in different low-resource settings and notably reduce the latency compared to the generation-based model.
- We conduct extensive experiments in different few-shot settings and explore the combination of cross-lingual and cross-domain scenarios.

2 Related Work

2.1 Task-Oriented Semantic Parsing

The majority of works on task-oriented semantic parsing focused on non-compositional user queries (Mesnil et al., 2013; Liu and Lane, 2016; Goo et al., 2018; Zhang et al., 2019), which turns the parsing task into a combination of intent detection and slot filling. Recently, Gupta et al. (2018)

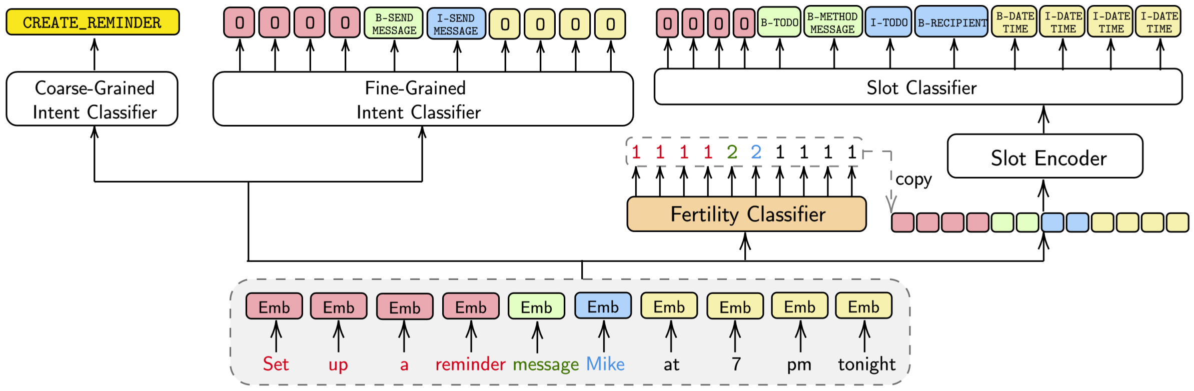

Figure 3: The architecture of X2Parser. We consider the TCSP task as a combination of the coarse-grained intent classification, fine-grained intent prediction, and slot filling tasks.

introduced a new dataset, called TOP, annotated with complex nested intents and slots and proposed to use the hierarchical representations to model the task. After that, Rongali et al. (2020) showed that leveraging a sequence-to-sequence model based on a copy mechanism (See et al., 2017) to directly generate the hierarchical representations was effective at parsing the nested queries. Taking this further, Chen et al. (2020) and Li et al. (2020a) extended the TOP dataset into multiple domains and multiple languages, and Li et al. (2020a) conducted zero-shot cross-lingual experiments using the combination of the multilingual pre-trained models (Conneau et al., 2020; Tran et al., 2020) and the copy mechanism method proposed in Rongali et al. (2020). Lately, Babu et al. (2021) and Shrivastava et al. (2021), which are concurrent works of X2Parser, proposed to tackled the TCSP task in a non-autoregressive way. Different from them, we propose to flatten the hierarchical representations and cast the problem into several sequence labeling tasks.

2.2 Language and Domain Adaptation

Recently, cross-lingual and cross-domain models that aim to tackle low-resource issues have been applied to natural language understanding (Conneau et al., 2018; Huang et al., 2019; Conneau et al., 2020; Gururangan et al., 2020), sentiment analysis (Zhou et al., 2016; Ziser and Reichart, 2017), task-oriented semantic parsing (Chen et al., 2018; Schuster et al., 2019; Liu et al., 2019; Wu et al., 2019; Liu et al., 2020a; Chen et al., 2020; Liu et al., 2020b), named entity recognition (Ni et al., 2017; Xie et al., 2018; Jia et al., 2019; Liu et al., 2020c), speech recognition (Mimura et al., 2017; Winata et al., 2020), abstractive summarization (Zhu et al., 2019; Ouyang et al., 2019; Yu et al., 2021), etc. Despite numerous studies related to the cross-lingual and cross-domain areas, only a few of them have explored how to effectively adapt models to the target languages in target domains, and the investigated tasks are limited to sentiment analysis (Fernández et al., 2016; Li et al., 2020b), abusive language detection (Pamungkas and Patti, 2019), and machine reading comprehension (Charlet et al., 2020). To the best of our knowledge, we are the first to study the combination of cross-lingual and cross-domain adaptations in the TCSP task.

3 Task Decomposition

In this section, we first introduce the intuition of decomposing the compositional semantic parsing into intent predictions and slot filling. Then, we describe how we construct intent and slot labels.

3.1 Intuition of Task Decomposition

We argue that hierarchical representations containing nested annotations for intents and slots are relatively complex. We need large enough training data to train a good model based on such representations, and the model's performance will be greatly limited in low-resource scenarios. Therefore, instead of incorporating intents and slots into one representation, we propose to predict them separately so that we can simplify the parsing problem and enable the model to easily learn the skills for each decomposed task, and finally, our model can achieve a better adaptation ability in low-resource scenarios. As illustrated in Figure 2, we obtain the coarse-grained intent, flattened fine-grained intents and flattened slot labels from the hierarchical representations, and train the model based on these three categories in a multi-task fashion. Note that we can always reconstruct the hierarchical representations based on the labels in these three categories, which means that the decomposed labels and the hierarchical labels are equivalent.

3.2 Label Constructions

Slot Labels We extract nested slot labels from the hierarchical representations and assign the labels to corresponding tokens based on the BIO (begin-inside-outside) structure. As we can see from Figure 2, there could exist multiple slot labels for one token, and we consider the order of the labels so as to reconstruct the hierarchical representations. Specifically, we put the more fine-grained slot label at the later position. For example, "message" (in Figure 2) has `B-TODO` and `B-METHOD-MESSAGE` labels, and `B-METHOD-MESSAGE` comes after `B-TODO` since it is a more fine-grained slot label.

Intent Labels Each data sample has one intent label for the whole user utterance, and we extract it as an individual coarse-grained intent label. For the intents expressed by partial tokens (i.e., fine-grained intents), we use the BIO structure to label the corresponding tokens. We notice that we only need to assign one intent label to each token since the nested cases for intents are relatively simple.[3] Therefore, the fine-grained intent classification becomes a sequence labeling task.

4 X2Parser

The model architecture of our X2Parser is illustrated in Figure 3. To enable the cross-lingual ability of our model, we leverage the multilingual pre-trained model XLM-R (Conneau et al., 2020) as the sequence encoder. Let us define $X = \{x_1, x_2, ..., x_n\}$ as the user utterance and $H = \{h_1, h_2, ..., h_n\}$ as the hidden states (denoted as Emb in Figure 3) from XLM-R.

4.1 Slot Predictor

The slot predictor consists of a fertility classifier, a slot encoder, and a slot classifier. Inspired by Gu et al. (2018), the fertility classifier learns to predict the number of slot labels for each token, and then it copies the corresponding number of hidden states. Finally, the slot classifier is trained to conduct the sequence labeling based on the slot labels we constructed. The fertility classifier not only helps the model identify the number of labels for each token but also guides the model to implicitly learn the nested slot information in user queries. It relieves the burden of the slot classifier, which needs to predict multiple slot entities for certain tokens.

Fertility Classifier (FC) We add a linear layer (FC) on top of the hidden states from XLM-R to predict the number of labels (fertility), which we formulate as follows:

$$F = \{f_1, f_2, ..., f_n\} = \text{FC}(\{h_1, h_2, ..., h_n\}), \tag{1}$$

where FC is an n-way classifier (n is the maximum label number) and $f_i (i \in [1, n])$ is a positive integer representing the number of labels for x_i.

Slot Filling After obtaining the fertility predictions, we copy the corresponding number of hidden states from XLM-R:

$$H' = \text{CopyHiddens}(H, F). \tag{2}$$

Then, we add a transformer encoder (Vaswani et al., 2017) (slot encoder (SE)) on top of H' to incorporate the sequential information into the hidden states, followed by adding a linear layer (slot classifier (SC)) to predict the slots, which we formulate as follows:

$$P_{\text{slot}} = \text{SC}(\text{SE}(H')), \tag{3}$$

where P_{slot} is a sequence of slots that has the same length as the sum of the fertility numbers.

4.2 Intent Predictor

Coarse-Grained Intent The coarse-grained intent is predicted based on the hidden state of the "`[CLS]`" token from XLM-R since it can be the representation for the whole sequence, and then we add a linear layer (coarse-grained intent classifier (CGIC)) on top of the hidden state to predict the coarse-grained intent:

$$p_{\text{cg}} = \text{CGIC}(h_{\text{cls}}), \tag{4}$$

where p_{cg} is a single intent prediction.

Fine-Grained Intent We add a linear layer (fine-grained intent classifier (FGIC)) on top of the hidden states H to produce the fine-grained intents:

$$P_{\text{fg}} = \text{FGIC}(\{h_1, h_2, ..., h_n\}), \tag{5}$$

where P_{fg} is a sequence of intent labels that has the same length as the input sequence.

[3] We place more details about how we construct labels for fine-grained nested intents in the Appendix A.

Model	en	es	fr	de	hi	th	Avg.
Seq2Seq w/ CRISS (Li et al., 2020a)	**84.20**	48.60	46.60	36.10	31.20	0.00	32.50
Seq2Seq w/ XLM-R (Li et al., 2020a)	83.90	50.30	43.90	42.30	30.90	26.70	38.82
Neural Layered Model (NLM)	82.40	59.99	58.16	54.91	29.31	28.78	46.23
X2Parser	83.39	**60.30**	**58.34**	**56.16**	**37.06**	**29.35**	**48.24**

Table 1: Exact match accuracies for the zero-shot **cross-lingual setting**. "Avg." denotes the averaged performance over all target languages (English excluded). The results of X2Parser and NLM are averaged over five runs.

Model	Alarm	Call.	Event	Msg.	Music	News	People	Recipe	Remind	Timer	Weather	Avg.
Seq2Seq	67.94	64.25	61.93	50.11	32.20	43.20	52.54	34.21	46.32	44.83	73.58	51.92
NLM	76.32	70.02	73.60	70.58	**56.52**	58.01	67.33	50.01	57.28	64.37	80.15	65.83
X2Parser	**76.72**	**73.16**	**77.33**	**71.45**	55.19	**64.43**	**69.77**	**51.78**	**58.86**	**65.98**	**81.17**	**67.80**

Table 2: Exact match accuracies (averaged over three runs) for the **cross-domain setting** in English. The scores represent the performance for the corresponding target domains. We use 10% of training samples in the target domain. **"Seq2Seq" denotes the "Seq2Seq w/ XLM-R" baseline** (same for the following tables and figures).

5 Experiments

5.1 Experimental Setup

Dataset We conduct the experiments on the MTOP dataset proposed by Li et al. (2020a), which contains six languages: English (en), German (de), French (fr), Spanish (es), and Thai (th), and 11 domains: alarm, calling, event, messaging, music, news, people, recipes, reminder, timer, and weather. The data statistics are reported in the Appendix B.

Cross-Lingual Setting In the cross-lingual setting, we use English as the source language and the other languages as target languages. In addition, we consider a zero-shot scenario where we only use English data for training.

Cross-Domain Setting In the cross-domain setting, we only consider training and evaluation in English. We choose ten domains as source domains and the other domain as the target domain. Different from the cross-lingual setting, we consider a few-shot scenario where we first train the model using the data from the ten source domains, and then we fine-tune the model using a few data samples (e.g., 10% of the data) from the target domain. We consider the few-shot scenario because zero-shot adapting the model to the target domain is extremely difficult due to the unseen intent and slot types, while zero-shot to target languages is easier using multilingual pre-trained models.

Cross-Lingual Cross-Domain Setting This setting combines the cross-lingual and cross-domain settings. Specifically, we first train the model on the English data from the ten source domains, and then fine-tune it on a few English data samples from the other (target) domain. Finally, we conduct the zero-shot evaluation on all the target languages of the target domain.

5.2 Baselines

Seq2Seq w/ XLM-R Rongali et al. (2020) proposed a sequence-to-sequence (Seq2Seq) model using a pointer-generator network (See et al., 2017) to handle nested queries, and achieved new state-of-the-art results in English. Li et al. (2020a) adopted this architecture for zero-shot cross-lingual adaptation. They replaced the encoder with the XLM-R (Conneau et al., 2020) and used a customized decoder to learn to generate intent and label types and copy tokens from the inputs.[4]

Seq2Seq w/ CRISS It is the same architecture as *Seq2Seq w/ XLM-R*, except that Li et al. (2020a) replaced XLM-R with the multilingual pre-trained model, CRISS (Tran et al., 2020), as the encoder for the zero-shot cross-lingual adaptation.

Neural Layered Model (NLM) This baseline conducts the multi-task training based on the same task decomposition as X2Parser, but it replaces the slot predictor module in X2Parser with a neural

[4] In order to compare the performance in the cross-domain and cross-lingual cross-domain settings, we follow Li et al. (2020a) to reimplement this baseline since the source code is not publicly available.

Model	Alarm	Call.	Event	Msg.	Music	News	People	Recipe	Remind	Timer	Weather	Avg.
Seq2Seq	34.29	47.00	41.81	25.86	19.21	25.39	22.13	16.12	9.80	20.01	36.90	22.25
NLM	48.53	43.30	44.62	43.32	36.25	28.60	43.29	28.54	20.50	34.16	59.57	39.15
X2Parser	**48.72**	**51.30**	**53.22**	**43.99**	**37.25**	**34.85**	**45.97**	**32.99**	**27.87**	**36.61**	**60.05**	**42.98**

Table 3: Exact match accuracies (averaged over three runs) for the **cross-lingual cross-domain setting**. The result for each domain is the averaged performance over all target languages. We use 10% of training samples in the English target domain, and do not use any data in the target languages.

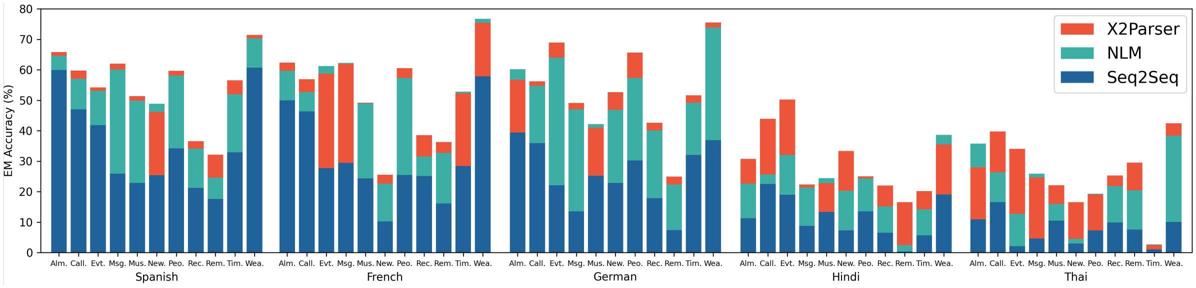

Figure 4: Full **cross-lingual cross-domain** results (across all target languages of target domains) for Table 3.

layered model (Ju et al., 2018),[5] while keeping the other modules the same. Unlike our fertility-based slot predictor, NLM uses several stacked layers to predict entities of different levels. We use this baseline to verify the effectiveness of our fertility-based slot predictor.

5.3 Training Details

We use XLM-R Large (Conneau et al., 2020) as the sequence encoder. For a word (in an utterance) with multiple subword tokens, we take the representations from the first subword token to predict the labels for this word. The transformer encoder (slot encoder) has one layer with a head number of 4, a hidden dimension of 400, and a filter size of 64. We set the fertility classifier as a 3-way classifier since the maximum label number for each token in the dataset is 3. We train X2Parser using the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 2e-5 and a batch size of 32. We follow Li et al. (2020a) and use the exact match accuracy to evaluate the models. For our model, the prediction is considered correct only when the predictions for the coarse-grained intent, fine-grained intents, and the slots are all correct. To ensure a fair comparison, we use the same three random seeds to run each model and calculate the averaged score for each target language and domain.

6 Results & Discussion

6.1 Main Results

Cross-Lingual Setting As we can see from Table 1, X2Parser achieves similar performance in English compared to Seq2Seq-based models, while it significantly outperforms them in the zero-shot cross-lingual setting, with $\sim$10% accuracy improvement on average. In the English training process, the Seq2Seq-based models can well learn the specific scope of tokens that need to be copied and assigned to a specific label type based on numerous training data. However, these models will easily lose effectiveness when the input sequences are in target languages due to the inherent variances across languages and the difficulty of generating hierarchical representations. X2Parser separates the TCSP task into predicting intents and slots individually, which lowers the task difficulty and boosts its zero-shot adaptation ability to target languages. Interestingly, we find that compared to *Seq2Seq w/ XLM-R*, X2Parser greatly boosts the performance on target languages that are topologically close to English (e.g., French (fr)) with more than 10% scores, while the improvements for languages that are topologically distant from English (e.g., Thai (th) and Hindi (hi)) are relatively limited. We argue that the large discrepancies between English and Thai make the representation alignment quality between English and Thai (Hindi) in XLM-R relatively low, and their different language patterns lead to unstable slot and intent predictions. These

[5] This model was originally proposed to tackle the nested named entity recognition task

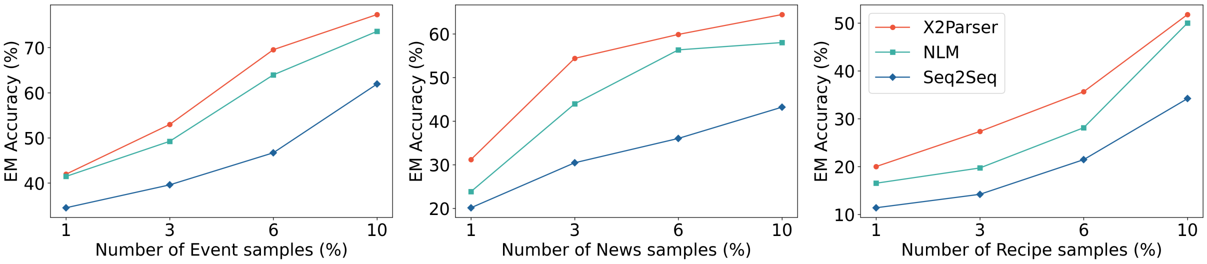

Figure 5: Few-shot exact match results on the **cross-domain setting** for Event, News and Recipe target domains.

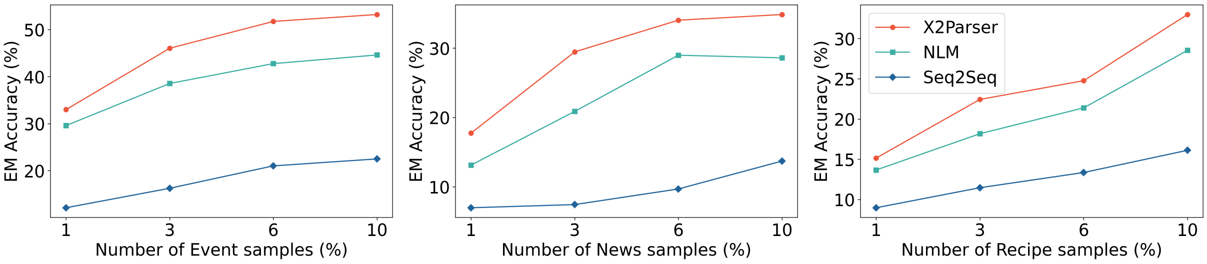

Figure 6: Few-shot exact match results on the **cross-lingual cross-domain setting** for Event, News and Recipe target domains. The results are averaged over all target languages.

factors limit the improvement for X2Parser on the adaptation to topologically distant languages.

From Table 1, although NLM achieves marginally lower performance in English compared to *Seq2Seq w/ XLM-R*, it produces significant improvements in target languages. This can be attributed to the fact that NLM leverages the same task decomposition as X2Parser, which further indicates the effectiveness of decomposing the TCSP task into intent and slot predictions for low-resource scenarios. Additionally, X2Parser surpasses NLM by ∼2% exact match accuracy on average in target languages. We conjecture that the stacked layers in NLM could make the model confused about which layer needs to generate which entity types, and this confusion is aggravated in the zero-shot cross-lingual setting where no training data are available. However, our fertility-based method helps the model implicitly learn the structure of hierarchical slots by predicting the number of labels for each token, which allows the slot classifier to predict the slot types more easily in the cross-lingual setting.

Cross-Domain Setting As shown in Table 2, X2Parser and NLM notably surpass the Seq2Seq model, with ∼15% improvements on the averaged scores. This can be largely attributed to the effectiveness of our proposed task decomposition for low-resource scenarios. Seq2Seq models need to learn when to generate the label, when to copy tokens from the inputs, and when to produce the end of the label to generate hierarchical representations. This generation process requires a relatively large number of data samples to learn, which leads to the weak few-shot cross-domain performance for the Seq2Seq model. Furthermore, X2Parser outperforms NLM, with a ∼2% averaged score. We conjecture that our fertility classifier guides the model to learn the inherent hierarchical information from the user queries, making it easier for the slot classifier to predict slot types for each token. However, the NLM's slot classifier, which consists of multiple stacked layers, needs to capture the hierarchical information and correctly assign slot labels of different levels to the corresponding stacked layer, which requires relatively larger data to learn.

Cross-Lingual Cross-Domain Setting From Table 3 and Figure 4, we can further observe the effectiveness of our proposed task decomposition and X2Parser in the cross-lingual cross-domain setting. X2Parser and NLM consistently outperform the Seq2Seq model in all target languages of the target domains and boost the averaged exact match accuracy by ∼20%. Additionally, from Table 3, X2Parser also consistently outperforms NLM on all 11 domains and surpasses it by 3.84% accuracy on average. From Figure 4, X2Parser greatly improves on NLM in topologically distant languages (i.e., Hindi and Thai). It illustrates the powerful transferability and robustness of the fertility-based

Model	Spanish		French		German		Hindi		Thai		Average	
	NN	Nested	NN	Nested	NN	Nested	NN	Nested	NN	Nested	NN	Nested
Seq2Seq	56.21	29.38	48.11	32.83	46.02	20.25	37.84	22.30	33.27	13.56	44.29	23.66
NLM	65.65	**41.95**	61.02	42.91	56.90	37.94	36.48	24.36	34.15	15.70	50.84	32.57
X2Parser	**66.69**	39.19	**63.45**	**44.28**	**58.43**	**39.71**	**42.64**	**28.55**	**35.96**	**16.67**	**53.43**	**33.68**

Table 4: Zero-shot cross-lingual exact match accuracies for nested and non-nested (NN) cases.

slot prediction that enables X2Parser to have a good zero-shot cross-lingual performance after it is fine-tuned to the target domain.

6.2 Few-shot Analysis

We conduct few-shot experiments using different sample sizes from the target domain for the cross-domain and cross-lingual cross-domain settings. The few-shot results on the Event, News, and Recipe target domains for both settings[6] are shown in Figure 5 and Figure 6. We find that the performance of the Seq2Seq model is generally poor in both settings, especially when only 1% of data samples are available. With the help of the task decomposition, NLM and X2Parser remarkably outperform the Seq2Seq model in various target domains for both the cross-domain and cross-lingual cross-domain settings across different few-shot scenarios (from 1% to 10%). Moreover, X2Parser consistently surpasses NLM for both the cross-domain and cross-lingual cross-domain settings in different few-shot scenarios, which further verifies the strong adaptation ability of our model.

Interestingly, we observe that the improvement of X2Parser over Seq2Seq grows as the number of training samples increases. For example, in the cross-lingual cross-domain setting of the event domain, the improvement goes from 20% to 30% as the training data increases from 1% to 10%. We hypothesize that in the low-resource scenario, the effectiveness of X2Parser will be greatly boosted when a relatively large number of data samples are available, while the Seq2Seq model needs much larger training data to achieve good performance.

6.3 Analysis on Nested & Non-Nested Data

To further understand how our model improves the performance, we split the test data in the MTOP dataset (Li et al., 2020a) into nested and non-nested samples. We consider the user utterances that do not have fine-grained intents and nested slots as the non-nested data sample and the rest of the data as the nested data sample. As we can see from Table 4, X2Parser significantly outperforms the Seq2Seq model on both nested and non-nested user queries with an average of ∼10% accuracy improvement in both cases. In addition, X2Parser also consistently surpasses NLM on all target languages in both the nested and non-nested scenarios, except for the Spanish nested case, which further illustrates the stable and robust adaptation ability of X2Parser.

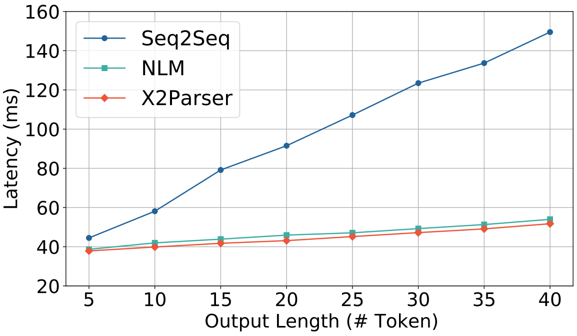

Figure 7: Averaged latencies for our model and baselines on different output lengths of the MTOP dataset.

6.4 Latency Analysis

We can see from Figure 7 that, as the output length increases, the latency discrepancy between the Seq2Seq-based model (Seq2Seq) and sequence labeling-based models (NLM and X2Parser) becomes larger, and when the output length reaches 40 tokens (around the maximum length in MTOP), X2Parser can achieve an up to 66% reduction in latency compared to the Seq2Seq model. This can be attributed to the fact that the Seq2Seq model has to generate the outputs token by token, while X2Parser and NLM can directly generate all the outputs. In addition, the inference speed of X2Parser is slightly faster than that of NLM. This is because NLM uses several stacked layers to predict slot entities of different levels, and the higher-level layer has to wait for the predictions from the lower-level layer, which slightly decreases the inference speed.

[6] We only report three domains due to the page limit, and place the full results for all 11 target domains in the Appendix C and Appendix D.

7 Conclusion

In this paper, we develop a transferable and non-autoregressive model (X2Parser) for the TCSP task that can better adapt to target languages and domains with a faster inference speed. Unlike previous TCSP models that learn to generate hierarchical representations, we propose to decompose the task into intent and slot predictions so as to lower the difficulty of the task, and then we cast both prediction tasks into sequence labeling problems. After that, we further propose a fertility-based method to cope with the slot prediction task where each token could have multiple labels. Results illustrate that X2Parser significantly outperforms strong baselines in all low-resource settings. Furthermore, our model is able to reduce the latency by up to 66% compared to the generation-based model.

Acknowledgement

We want to say thanks to the anonymous reviewers for the insightful reviews and constructive feedback. This work is partially funded by ITF/319/16FP and MRP/055/18 of the Innovation Technology Commission, the Hong Kong SAR Government.

References

Arun Babu, Akshat Shrivastava, Armen Aghajanyan, Ahmed Aly, Angela Fan, and Marjan Ghazvininejad. 2021. Non-autoregressive semantic parsing for compositional task-oriented dialog. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2969–2978.

Delphine Charlet, Géraldine Damnati, Frédéric Béchet, Johannes Heinecke, et al. 2020. Cross-lingual and cross-domain evaluation of machine reading comprehension with squad and calor-quest corpora. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5491–5497.

Wenhu Chen, Jianshu Chen, Yu Su, Xin Wang, Dong Yu, Xifeng Yan, and William Yang Wang. 2018. Xl-nbt: A cross-lingual neural belief tracking framework. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 414–424.

Xilun Chen, Asish Ghoshal, Yashar Mehdad, Luke Zettlemoyer, and Sonal Gupta. 2020. Low-resource domain adaptation for compositional task-oriented semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5090–5100.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485.

Alejandro Moreo Fernández, Andrea Esuli, and Fabrizio Sebastiani. 2016. Distributional correspondence indexing for cross-lingual and cross-domain sentiment classification. *Journal of artificial intelligence research*, 55:131–163.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. 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 2485–2494.

Chen Jia, Xiaobo Liang, and Yue Zhang. 2019. Cross-domain ner using cross-domain language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2464–2474.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2020a. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. *arXiv preprint arXiv:2008.09335*.

Juntao Li, Ruidan He, Hai Ye, Hwee Tou Ng, Lidong Bing, and Rui Yan. 2020b. Unsupervised domain adaptation of a pretrained cross-lingual language model. In *IJCAI*.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *Interspeech 2016*, pages 685–689.

Zihan Liu, Jamin Shin, Yan Xu, Genta Indra Winata, Peng Xu, Andrea Madotto, and Pascale Fung. 2019. Zero-shot cross-lingual dialogue systems with transferable latent variables. 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 1297–1303.

Zihan Liu, Genta Indra Winata, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2020a. Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8433–8440.

Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. 2020b. Coach: A coarse-to-fine approach for cross-domain slot filling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 19–25.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2020c. Crossner: Evaluating cross-domain named entity recognition. *arXiv preprint arXiv:2012.04373*.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *Interspeech*, pages 3771–3775.

Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara. 2017. Cross-domain speech recognition using nonparallel corpora with cycle-consistent adversarial networks. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 134–140. IEEE.

Jian Ni, Georgiana Dinu, and Radu Florian. 2017. Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1470–1480.

Jessica Ouyang, Boya Song, and Kathleen McKeown. 2019. A robust abstractive system for cross-lingual summarization. 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 2025–2031.

Endang Wahyu Pamungkas and Viviana Patti. 2019. Cross-domain and cross-lingual abusive language detection: A hybrid approach with deep learning and a multilingual lexicon. In *Proceedings of the 57th annual meeting of the association for computational linguistics: Student research workshop*, pages 363–370.

Subendhu Rongali, Luca Soldaini, Emilio Monti, and Wael Hamza. 2020. Don't parse, generate! a sequence to sequence architecture for task-oriented semantic parsing. In *Proceedings of The Web Conference 2020*, pages 2962–2968.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3795–3805.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Akshat Shrivastava, Pierce Chuang, Arun Babu, Shrey Desai, Abhinav Arora, Alexander Zotov, and Ahmed Aly. 2021. Span pointer networks for non-autoregressive task-oriented semantic parsing. *arXiv preprint arXiv:2104.07275*.

Chau Tran, Yuqing Tang, Xian Li, and Jiatao Gu. 2020. Cross-lingual retrieval for iterative self-supervised training. In *Advances in Neural Information Processing Systems*, volume 33, pages 2207–2219. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, Peng Xu, and Pascale Fung. 2020. Learning fast adaptation on cross-accented speech recognition. *Proc. Interspeech 2020*, pages 1276–1280.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.

Jiateng Xie, Zhilin Yang, Graham Neubig, Noah A Smith, and Jaime G Carbonell. 2018. Neural cross-lingual named entity recognition with minimal resources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 369–379.

Tiezheng Yu, Zihan Liu, and Pascale Fung. 2021. Adaptsum: Towards low-resource domain adaptation for abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5892–5904.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and S Yu Philip. 2019. Joint slot filling and intent detection via capsule neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5259–5267.

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Attention-based lstm network for cross-lingual sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 247–256.

Junnan Zhu, Qian Wang, Yining Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, and Chengqing Zong. 2019. Ncls: Neural cross-lingual summarization. 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 3045–3055.

Yftah Ziser and Roi Reichart. 2017. Neural structural correspondence learning for domain adaptation. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 400–410.

A Intent Label Construction

In this section, we further describe how we convert the fine-grained intent prediction into a sequence labeling task (each token has only one label). We use a few examples to illustrate our intent label construction method.

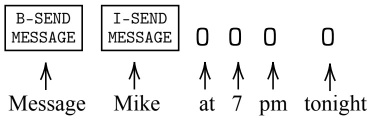

Figure 8: A labeling example for non-nested intent.

As illustrated in Figure 8, when there are no nested intents in the input utterance, we follow the BIO structure to give intent labels.

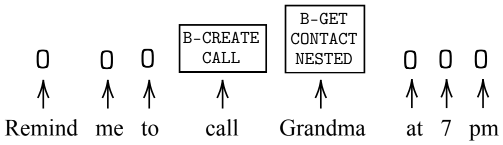

Figure 9: A labeling example for nested intent.

We can see from Figure 9 that "call Grandma" is a CREATE-CALL intent and "Grandma" is a GET-CONTACT intent. Hence, the GET-CONTACT intent is nested in the CREATE-CALL intent. We use a special intent label (with "NESTED") for the "GET-CONTACT" intent (B-GET-CONTACT-NESTED) to represent that this intent is nested in another intent, and hence, the scope of the CREATE-CALL intent is automatically expanded from "call" to "call Grandma". [7]

Note that we cannot apply this labeling method to the slot prediction since one token in the user utterance could be the starting token for more than one slot entity. If that is the case, we have to use more than one slot label for this token to denote the starting position for each slot entity. Given that in the MTOP dataset, one token will not be the starting token of more than one intent, we can apply this method for the intent label construction. In the future, when more complex and sophisticated datasets are collected for the task-oriented compositional semantic parsing task, where there could exist more than one intent label for each token, we can always use the fertility-based method (currently applied for the slot prediction) for the intent prediction.

B Data Statistics

The data statistics for MTOP are shown in Table 5.

C Few-shot Cross-Domain Results

Full few-shot cross-domain results across all 11 target domains are shown in Figure 10 and Table 6.

D Few-shot Cross-Lingual Cross-Domain Results

Full few-shot cross-lingual cross-domain results across all 11 target domains are shown in Figure 11 and Tables 7, 8, 9, 10, and 11.

[7] We notice that if two intents have overlaps, one intent either fully covers the other intent or is fully covered by the other intent.

Domain	Number of Utterances						Intent	Slot
	English	German	French	Spanish	Hindi	Thai	Types	Types
Alarm	1,783	1,581	1,706	1,377	1,510	1,783	6	5
Calling	2,872	2,797	2,057	2,515	2,490	2,872	19	14
Event	1,081	1,051	1,115	911	988	1,081	12	12
Messaging	1,053	1,239	1,335	1,164	1,082	1,053	7	15
Music	1,648	1,499	1,312	1,509	1,418	1,648	27	12
News	1,393	905	1,052	1,130	930	1,393	3	6
People	1,449	1,392	763	1,408	1,168	1,449	17	16
Recipes	1,586	1,002	762	1,382	929	1,586	3	18
Reminder	2,439	2,321	2,202	1,811	1,833	2,439	19	17
Timer	1,358	1,014	1,165	1,159	1,047	1,358	9	5
Weather	2,126	1,785	1,990	1,816	1,800	2,126	4	4
Total	18,788	16,585	15,459	16,182	15,195	18,788	117	78

Table 5: Data statistics of the MTOP dataset. The data are roughly divided into a 70:10:20 percent split for train, eval and test

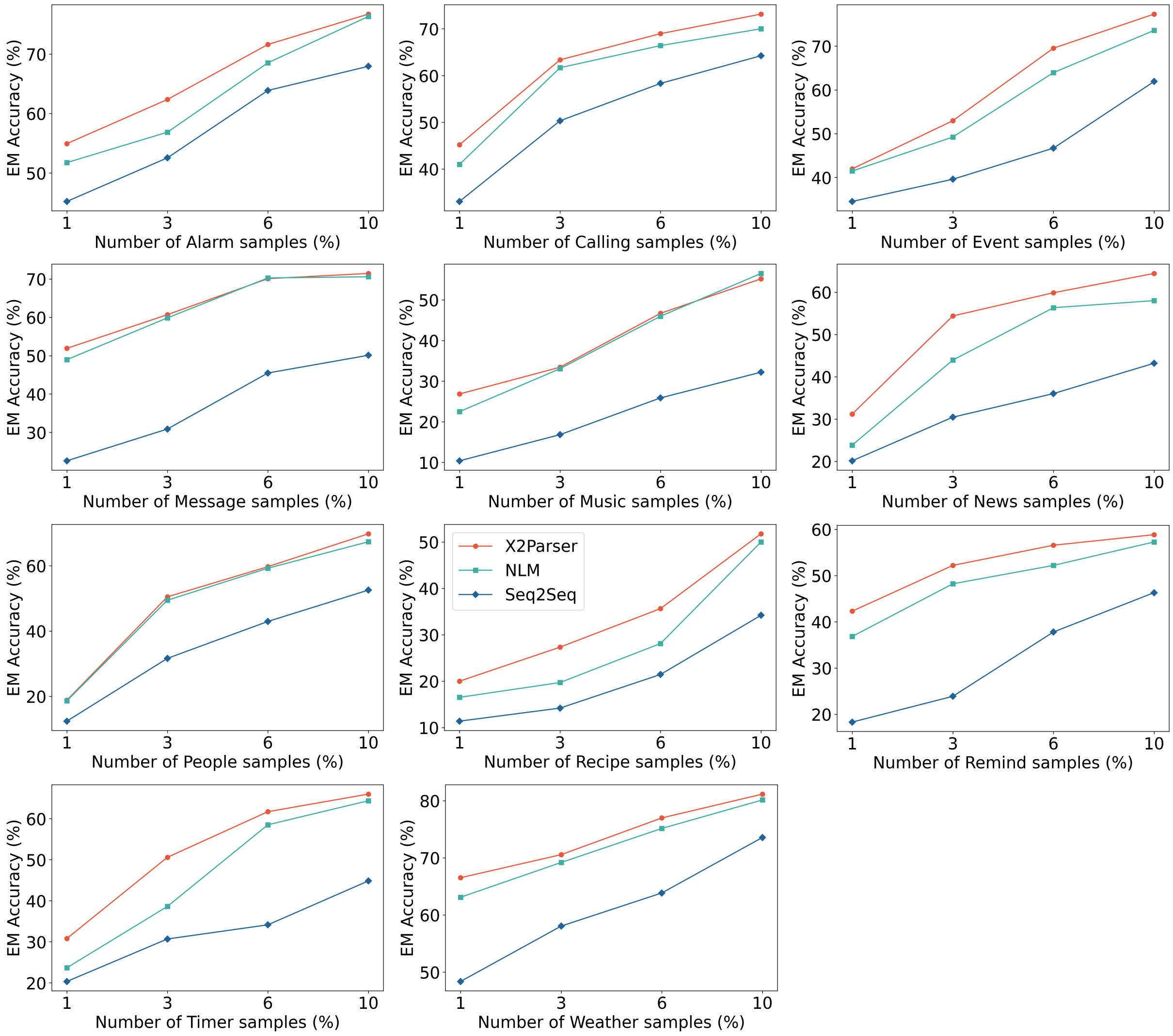

Figure 10: Few-shot exact match accuracies for the **cross-domain setting** across all 11 target domains.

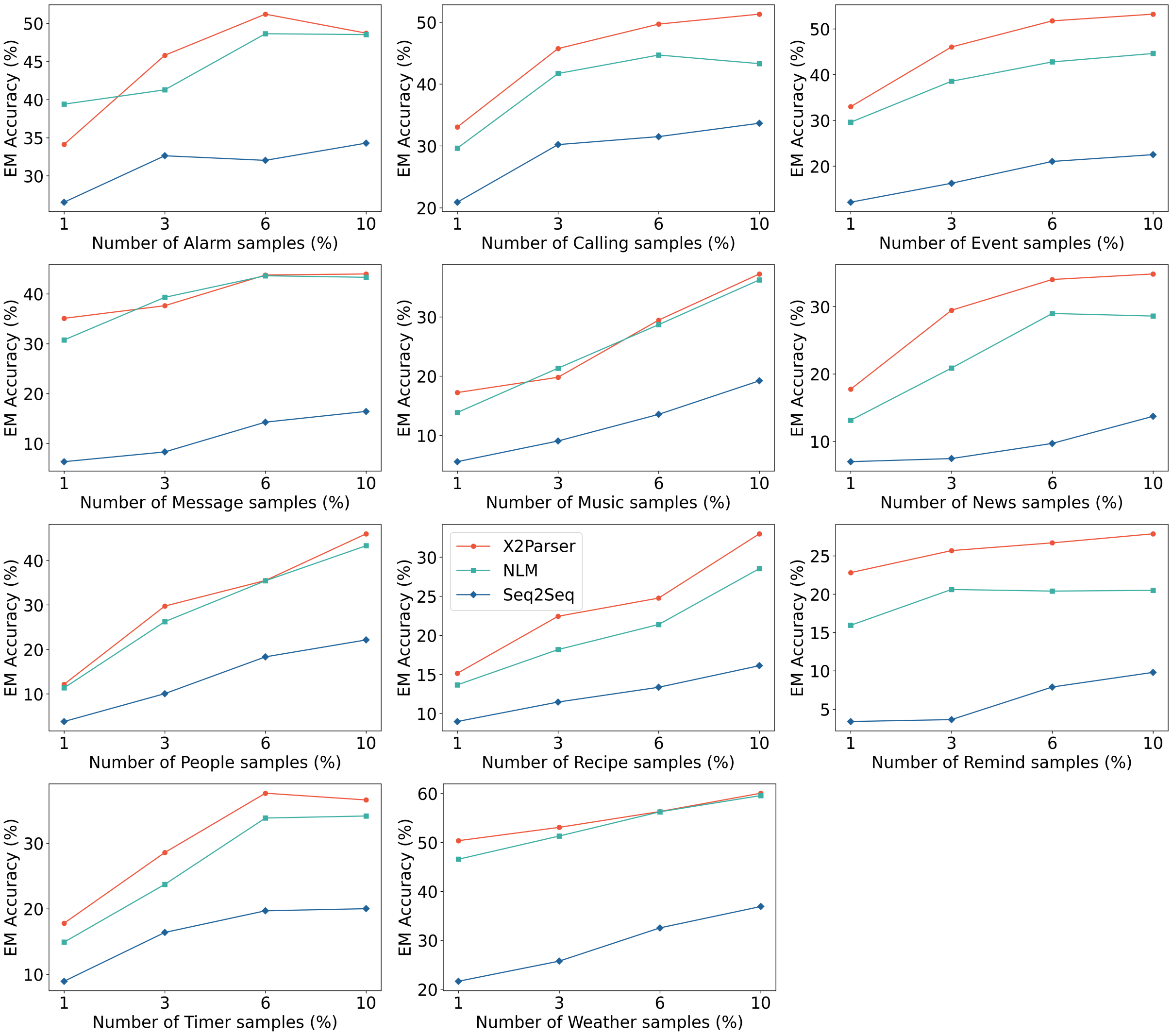

Figure 11: Few-shot Exact match accuracies for the **cross-lingual cross-domain setting** across all 11 target domains. The results are averaged over all target languages.

# Sample	Model	Alarm	Call.	Event	Msg.	Music	News	People	Recipe	Remind	Timer	Weather	Avg
1%	Seq2Seq	45.22	33.07	34.52	22.58	10.38	20.14	12.43	11.39	18.33	20.34	48.36	25.16
	NLM	51.75	41.00	41.46	48.97	22.49	23.84	18.65	16.52	36.84	23.67	63.11	35.30
	X2Parser	**54.94**	**45.20**	**41.96**	**51.91**	**26.83**	**31.19**	**18.83**	**20.00**	**42.31**	**30.80**	**66.53**	**39.14**
3%	Seq2Seq	52.55	50.33	39.59	30.87	16.82	30.45	31.64	14.20	23.90	30.69	58.06	34.46
	NLM	56.86	61.68	49.24	59.86	33.06	43.95	49.43	19.71	48.23	38.62	69.20	48.17
	X2Parser	**62.36**	**63.37**	**52.97**	**60.70**	**33.42**	**54.38**	**50.47**	**27.34**	**52.21**	**50.58**	**70.57**	**52.58**
6%	Seq2Seq	63.88	58.32	46.70	45.48	25.87	36.03	42.94	21.45	37.81	34.14	63.86	43.32
	NLM	68.53	66.42	63.96	**70.28**	45.98	56.33	59.23	28.12	52.21	58.51	75.16	58.61
	X2Parser	**71.61**	**68.97**	**69.54**	70.09	**46.70**	**59.87**	**59.70**	**35.65**	**56.57**	**61.70**	**77.00**	**61.58**
10%	Seq2Seq	67.94	64.25	61.93	50.11	32.20	43.20	52.54	34.21	46.32	44.83	73.58	51.92
	NLM	76.32	70.02	73.60	70.58	**56.52**	58.01	67.33	50.01	57.28	64.37	80.15	65.83
	X2Parser	**76.72**	**73.16**	**77.33**	**71.45**	55.19	**64.43**	**69.77**	**51.78**	**58.86**	**65.98**	**81.17**	**67.80**

Table 6: Complete results of the **cross-domain setting**.

# Sample	Model	Alarm	Call.	Event	Msg.	Music	News	People	Recipe	Remind	Timer	Weather	Avg
1%	Seq2Seq	33.81	28.00	24.29	9.89	8.22	13.59	4.49	15.87	7.87	14.29	38.86	18.11
	NLM	44.41	31.75	36.53	41.95	14.36	16.34	12.82	**23.28**	21.76	24.67	57.92	29.62
	X2Parser	**51.51**	**36.67**	**36.72**	**51.84**	**20.86**	**19.90**	**15.60**	19.05	**26.16**	**30.74**	**59.65**	**33.52**
3%	Seq2Seq	48.58	41.75	30.51	14.07	12.35	10.68	17.31	19.84	6.94	28.57	42.82	24.86
	NLM	53.41	50.33	43.31	**54.37**	**26.10**	29.45	34.19	**24.61**	25.46	38.96	**64.03**	40.38
	X2Parser	**56.06**	**54.75**	**46.14**	53.23	24.25	**33.82**	**34.61**	23.54	**30.79**	**44.73**	63.61	**42.32**
6%	Seq2Seq	51.70	43.50	36.16	25.48	16.91	18.45	27.56	23.02	12.50	33.33	51.49	30.92
	NLM	60.32	52.83	48.02	60.84	**41.46**	**47.25**	**46.58**	26.19	27.70	53.10	67.41	48.34
	X2Parser	**66.10**	**61.67**	**53.30**	**61.85**	40.24	44.82	44.01	**27.78**	**31.48**	**53.97**	**68.81**	**50.37**
10%	Seq2Seq	59.94	47.00	41.81	25.86	22.85	25.39	34.21	21.25	17.59	32.90	60.69	35.41
	NLM	64.57	57.08	53.11	60.08	49.86	**48.84**	58.12	34.13	24.61	51.95	70.30	52.06
	X2Parser	**65.81**	**59.75**	**54.24**	**61.98**	**51.36**	46.12	**59.62**	**36.51**	**32.10**	**56.57**	**71.45**	**54.14**

Table 7: Complete results of the **cross-lingual cross-domain setting** in Spanish.

# Sample	Model	Alarm	Call.	Event	Msg.	Music	News	People	Recipe	Remind	Timer	Weather	Avg
1%	Seq2Seq	43.94	30.77	15.72	11.55	8.07	5.66	8.09	12.15	5.87	13.16	35.67	17.33
	NLM	53.13	35.04	**41.51**	46.75	16.07	7.55	15.32	18.78	26.01	23.51	59.74	31.22
	X2Parser	**54.65**	**37.48**	**41.51**	**49.40**	**21.67**	**10.27**	**18.01**	**20.44**	**29.28**	**26.67**	**65.07**	**34.04**
3%	Seq2Seq	51.21	42.91	16.98	10.76	9.07	8.81	8.09	14.92	5.22	22.11	43.54	21.24
	NLM	55.66	49.58	51.57	**54.98**	**25.32**	17.82	33.34	23.94	30.73	34.91	65.73	40.33
	X2Parser	**59.70**	**52.87**	**54.72**	52.99	24.40	**18.03**	**38.60**	**27.81**	**31.74**	**40.18**	**65.92**	**42.45**
6%	Seq2Seq	49.70	45.24	25.79	19.92	17.86	5.66	19.85	19.89	14.78	31.05	55.62	27.76
	NLM	64.08	50.00	57.86	**63.88**	36.67	23.27	**48.41**	28.73	28.48	50.18	**74.72**	47.84
	X2Parser	**66.77**	**59.22**	**58.49**	59.49	**38.45**	**25.16**	46.81	**29.84**	**35.29**	**55.62**	71.82	**49.72**
10%	Seq2Seq	50.00	46.34	27.67	29.48	24.29	10.18	25.43	25.10	16.09	28.42	57.81	30.98
	NLM	59.64	52.68	**61.22**	**62.29**	48.93	22.59	57.35	31.49	32.75	**52.81**	**76.69**	50.77
	X2Parser	**62.32**	**56.84**	58.70	61.89	**49.17**	**25.58**	**60.54**	**38.49**	**36.31**	52.28	75.37	**52.50**

Table 8: Complete results of the **cross-lingual cross-domain setting** in French.

# Sample	Model	Alarm	Call.	Event	Msg.	Music	News	People	Recipe	Remind	Timer	Weather	Avg
1%	Seq2Seq	35.13	21.61	7.36	8.81	7.98	11.74	3.58	6.90	0.89	17.19	24.15	13.21
	NLM	46.27	37.43	44.58	27.81	18.95	24.50	14.70	11.84	16.03	22.00	57.86	29.27
	X2Parser	**52.03**	**39.32**	**45.40**	**34.72**	**21.68**	**33.97**	**15.17**	**15.29**	**19.98**	**27.09**	**60.52**	**33.20**
3%	Seq2Seq	38.81	35.45	15.34	8.81	13.89	13.26	15.41	11.03	2.46	28.12	23.69	18.75
	NLM	52.50	48.44	57.87	38.86	**28.94**	38.38	38.35	16.32	21.55	36.98	64.84	40.28
	X2Parser	**52.69**	**50.18**	**58.28**	**39.90**	25.82	**48.36**	**45.04**	**24.71**	**23.64**	**39.32**	**65.83**	**43.07**
6%	Seq2Seq	37.68	38.10	20.25	17.62	15.80	18.18	25.81	13.10	6.49	30.47	36.67	23.65
	NLM	54.77	50.00	62.78	42.14	34.75	47.35	50.54	21.72	21.92	47.27	70.31	45.78
	X2Parser	**59.39**	**55.31**	**69.32**	**48.70**	**35.08**	**51.77**	**53.53**	**28.74**	**25.28**	**50.00**	**73.20**	**50.03**
10%	Seq2Seq	39.38	35.90	22.09	13.47	25.17	22.83	30.24	17.83	7.38	32.03	36.89	25.75
	NLM	**60.13**	54.76	64.01	46.98	**42.16**	46.84	57.35	40.12	22.30	49.22	73.88	50.70
	X2Parser	56.75	**56.23**	**68.92**	**49.05**	40.96	**52.65**	**65.59**	**42.64**	**24.91**	**51.56**	**75.47**	**53.16**

Table 9: Complete results of the **cross-lingual cross-domain setting** in German.

# Sample	**Model**	**Alarm**	**Call.**	**Event**	**Msg.**	**Music**	**News**	**People**	**Recipe**	**Remind**	**Timer**	**Weather**	**Avg**
1%	Seq2Seq	11.99	13.30	9.85	1.46	1.65	3.23	1.91	3.08	0.00	0.00	6.89	4.85
	NLM	**16.10**	17.67	15.15	20.39	9.92	12.73	**9.54**	5.73	1.27	**3.25**	30.44	12.92
	X2Parser	**16.10**	**25.04**	**25.00**	**20.55**	**10.60**	**15.59**	7.76	**7.64**	**12.88**	**3.25**	**31.63**	**16.00**
3%	Seq2Seq	11.61	18.00	14.39	6.80	4.83	3.76	5.73	3.96	0.38	2.56	14.07	7.83
	NLM	14.23	28.62	23.49	**24.27**	13.54	14.87	16.29	9.25	3.28	5.47	**34.33**	17.06
	X2Parser	**25.59**	**34.76**	**40.91**	21.85	**14.46**	**34.77**	**19.08**	**17.03**	**15.66**	**16.93**	32.64	**24.88**
6%	Seq2Seq	14.61	13.73	18.18	6.31	10.26	4.84	11.83	5.29	0.38	2.56	12.87	9.17
	NLM	**29.63**	**35.29**	29.80	22.17	**20.11**	21.33	20.74	9.84	2.15	16.07	**38.42**	22.32
	X2Parser	27.97	34.05	**47.47**	**23.62**	17.36	**36.74**	**21.88**	**16.59**	**16.54**	**26.84**	33.34	**27.49**
10%	Seq2Seq	11.24	22.53	18.94	8.74	13.31	7.23	13.54	6.52	0.38	5.64	19.07	11.56
	NLM	22.58	25.61	32.07	21.36	**24.38**	20.25	24.30	15.13	2.40	14.19	**38.61**	21.90
	X2Parser	**30.71**	**43.92**	**50.25**	**22.33**	22.73	**33.33**	**25.06**	**22.02**	**16.54**	**20.17**	35.53	**29.33**

Table 10: Complete results of the **cross-lingual cross-domain setting** in Hindi.

# Sample	**Model**	**Alarm**	**Call.**	**Event**	**Msg.**	**Music**	**News**	**People**	**Recipe**	**Remind**	**Timer**	**Weather**	**Avg**
1%	Seq2Seq	7.82	10.81	3.40	0.00	1.82	0.64	0.94	6.90	2.31	0.00	2.59	3.38
	NLM	**37.08**	26.27	10.20	16.92	9.93	4.46	**4.40**	8.74	14.65	1.06	26.82	14.59
	X2Parser	34.12	**26.77**	**16.33**	**18.95**	**11.29**	**8.92**	4.25	**13.33**	**25.76**	**1.24**	**34.87**	**17.80**
3%	Seq2Seq	12.93	12.96	4.08	1.02	5.08	0.64	3.77	7.59	3.20	0.53	4.60	5.13
	NLM	30.61	31.54	16.49	**24.03**	**12.75**	3.82	8.96	16.78	22.02	**2.29**	27.59	17.90
	X2Parser	**35.03**	**36.09**	**30.16**	20.14	10.07	**12.32**	**11.32**	**19.08**	**26.55**	1.77	**37.36**	**21.81**
6%	Seq2Seq	6.46	16.95	4.76	2.03	6.86	1.27	6.60	5.52	5.32	1.06	6.03	5.71
	NLM	34.39	35.31	15.42	**29.10**	10.48	5.73	10.85	20.46	21.73	**2.65**	30.36	19.68
	X2Parser	**35.83**	**38.21**	**30.16**	25.21	**16.19**	**11.68**	**11.16**	**20.92**	**24.87**	1.77	**34.36**	**22.76**
10%	Seq2Seq	10.88	16.53	2.04	4.57	10.45	2.91	7.25	9.90	7.56	1.06	10.02	7.56
	NLM	**35.75**	26.34	12.70	**25.89**	15.92	4.46	**19.34**	21.84	20.45	**2.65**	38.39	20.34
	X2Parser	28.00	**39.75**	**34.01**	24.70	**22.04**	**16.56**	19.02	**25.29**	**29.50**	2.47	**42.43**	**25.80**

Table 11: Complete results of the **cross-lingual cross-domain setting** in Thai.