CALICO : Conversational Agent Localization via Synthetic Data Generation

Andy Rosenbaum[1] andros@amazon.com
Pegah Kharazmi[1] pkkharaz@amazon.com
Ershad Banijamali[1] ebanijam@amazon.com
Lu Zeng[1] luzeng@amazon.com
Chris DiPersio[1] dipersio@amazon.com
Pan Wei[1] panwei@amazon.com
Gokmen Oz[1] ogokmen@amazon.de
Clement Chung[2] chungcle@amazon.com
Karolina Owczarzak[2] karowc@amazon.com
Fabian Triefenbach[2] triefen@amazon.de
Wael Hamza[2] waelhamz@amazon.com

Amazon, Alexa AI

Abstract

We present CALICO, a method to fine-tune Large Language Models (LLMs) to localize conversational agent training data from one language to another. For slots (named entities), CALICO supports three operations: verbatim copy, literal translation, and *localization*, i.e. generating slot values more appropriate in the target language, such as city and airport names located in countries where the language is spoken. Furthermore, we design an iterative filtering mechanism to discard noisy generated samples, which we show boosts the performance of the downstream conversational agent. To prove the effectiveness of CALICO, we build and release a new human-*localized* (HL) version of the MultiATIS++ travel information test set in 8 languages. Compared to the original human-translated (HT) version of the test set, we show that our new HL version is more challenging. We also show that CALICO out-performs state-of-the-art LINGUIST (which relies on literal slot translation out of context) both on the HT case, where CALICO generates more accurate slot translations, and on the HL case, where CALICO generates *localized* slots which are closer to the HL test set.

1 Introduction and Related Work

Conversational agents usually apply Intent Classification (IC) and Slot Tagging (ST) (also known as Named Entity Recognition or NER) to infer semantics from the text of an agent-directed request (Tur and De Mori, 2011). In order to support global user bases, these agents are often multilingual.

IC+ST training data is typically scarce, especially in multilingual settings. While Large Langauge Models (LLMs) can perform IC+ST from few examples (e.g. Parikh et al. (2023)), lightweight

[1]Individual Contributor
[2]Manager/Advisor

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

```
INPUT:
<language> German </language>
<domain> travelinfo </domain>
<intent> flight </intent>
<include>
 [1 translation( first class ) ] ,
 [2 localization( united airlines ) ] ,
 [3 localization( denver ) ] ,
 [4 localization( baltimore ) ] ,
 [5 translation( december seventeenth ) ]
</include>
<labels>
 [1=fare_class , [2=airline ,
 [3=fromloc.city_name ,
 [4=toloc.city_name , [5=date
</labels>
<examples>
i need a [1 first class ] ticket
  on [2 united airlines ]
  from [3 denver ]
  to [4 baltimore ]
  scheduled for [5 december seventeenth ]
</examples>

OUTPUTS:
1. ich brauche ein [1 erste klasse ] ticket
   mit [2 lufthansa ]
   von [3 hamburg ]
   nach [4 köln ]
   für den [5 siebzehnten dezember ]
```

Figure 1: CALICO follows instructions to either literally translate entities (`fare_class`, `date`) or to *localize* entities (`airline`, `from_city`, `to_city`).

models such as encoder-only Transformers (Chen et al., 2019; Xu et al., 2020) are still useful for cost- and latency-sensitive applications that support very high throughput.

Synthetic Data Generation (SDG) from Large Language Models (LLMs) has become a popular trend to address the data scarcity problem (Rosenbaum et al., 2023). SDG approaches relevant to the IC and ST tasks include back-translation, (Bannard and Callison-Burch, 2005; Sennrich et al., 2016; Edunov et al., 2018; Xie et al., 2020) paraphrasing (Kumar et al., 2020; Cho et al., 2019; Malandrakis et al., 2019; Jolly et al., 2020; Panda et al., 2021) word replacement (Zhang et al., 2020; Dai and Adel, 2020; Wei and Zou, 2019), and carrier phrase regeneration (Kumar et al., 2022). A related thread is in-context generation of multilingual semantic parsing data (Rosenbaum et al., 2022a) and multi-party dialogs (Chen et al., 2023).

In terms of *cross-lingual* SDG for IC+ST, Machine Translation with Slot Alignment (MT-SA) is a strong baseline (Xu et al., 2020), however the separate alignment step *a posteriori* introduces noise, which negatively impacts the quality of the generated data and the downstream task model.

Recently proposed LINGUIST (Rosenbaum et al., 2022b) avoids the alignment problem by first machine-translating the slot values (out-of-context entities like "new orleans" or "december sixteenth"), and then generating a slot-annotated utterance in the target language incorporating the machine-translated slot values. (An example of LINGUIST output is "`book a flight to [1 new orleans ] on [2 december sixteenth ]`", where `1` and `2` indicate slot labels `to_city` and `date` respectively.)

In this work, we propose CALICO for cross-lingual SDG of IC+ST training data, which resolves two important limitations of LINGUIST (see Figure 1):

(i) **Contextualized Slot Value Translation**: LINGUIST translates the slots *a priori* and out of context, which can lead to cascading errors, due to the translation model choosing the wrong grammatical form in morphologically inflected languages, or choosing the wrong semantic translation altogether.

(For example, "light" can be a noun, synonym of "lamp"; or an adjective, opposite of "heavy"; or a verb.) By contrast, CALICO translates the slot values and carrier phrase text jointly, while producing the same slot-annotated output format as LINGUIST to avoid the alignment problem of MT-SA.

(ii) **Slot Value Localization**: in real-world systems, users are more likely to ask for entities specific to their locale, e.g. in German booking flights to or from "köln" on "lufthansa" instead of just asking for English entities and their translations, like "denver" or a literal translation of "united airlines". CALICO introduces a `localization` operator which instructs the model to replace the value in the source language with a *localized* version of the slot while translating the rest of the text around it.

To demonstrate the utility of slot value *localization*, we create a new human-localized (HL) version of the MultiATIS++ test set in all 9 languages[1], and benchmark CALICO compared to LINGUIST on the 6 languages shared with our data generation model. We show that our HL test set is more challenging that the original human-translated (HT) test set. We also show that CALICO out-performs LINGUIST both on the original HT version, by producing more accurate slot translations with the full sentence context, and on the new HL version by producing more relevant training data with localized slot values like city and airport names.

Furthermore, we improve the process of selecting from among the n-best generated CALICO outputs: instead of taking the output with lowest perplexity, we design an Iterative Filtering Mechanism (IFM) inspired by data augmentation through weak supervision (Chen et al., 2022). We use the downstream task model (IC+ST encoder fine-tuned on real data plus selected CALICO-generated synthetic data) to *re-select from among the n-best outputs based on matching the intent and slots* requested in the prompt. We show that the IFM improves the final IC+ST performance on the MultiATIS++ test set.

In summary, our contributions are threefold: (1) We propose CALICO to localize IC+ST training data with controls to either copy, translate, or *localize* slot values; (2) we create a new version of the MultiATIS++ non-English test set, which includes updated text and annotation with human-*localized* slot values such as city and airport names, benchmark our models on it, and release the test set; and (3) we design an iterative filtering mechanism to select model generated data and show that it improves IC+ST performance on the MultiATIS++ test set (both original and human-localized versions) compared to selecting the output with lowest perplexity.

2 Methodology

Like LINGUIST, CALICO is a generative Large Language Model (LLM) fine-tuned on an instruction prompt to generate synthetic training data for IC+ST. CALICO takes inspiration from the LINGUIST prompt, and supports additional slot operations.

2.1 CALICO Prompt Design

The CALICO prompt (Figure 1) differs from LINGUIST by adding controls at the slot level for three operations: `unchanged`, indicating a verbatim copy (e.g. for flight numbers), literal `translation`, and `localization`, i.e. replacement with a value more appropriate in the target language.

The `translation` operation of CALICO improves the slot translation quality compared to out-of-context MT applied *a priori* to LINGUIST. For example, without any context, the word "second" in English could be reasonably translated to Spanish as either "*segundo*", "*segunda*", "*segundos*", or "*segundas*", depending on the plurality and grammatical gender of the Spanish noun it modifies. Furthermore, if "second" is part of the phrase "the second of september", then it should be translated as "*dos*", meaning "two". CALICO attends to the entire input English sentence when generating translated values, and therefore can disambiguate such cases.

2.2 Training the CALICO Model

Similar to LINGUIST, we fine-tune CALICO from AlexaTM 5B seq2seq on cross-lingual prompts extracted from MASSIVE (FitzGerald et al., 2022b). See details in Section 3.2.1. Models are finetuned for 10 epochs using a batch size of 16.

[1] https://github.com/amazon-science/matispp

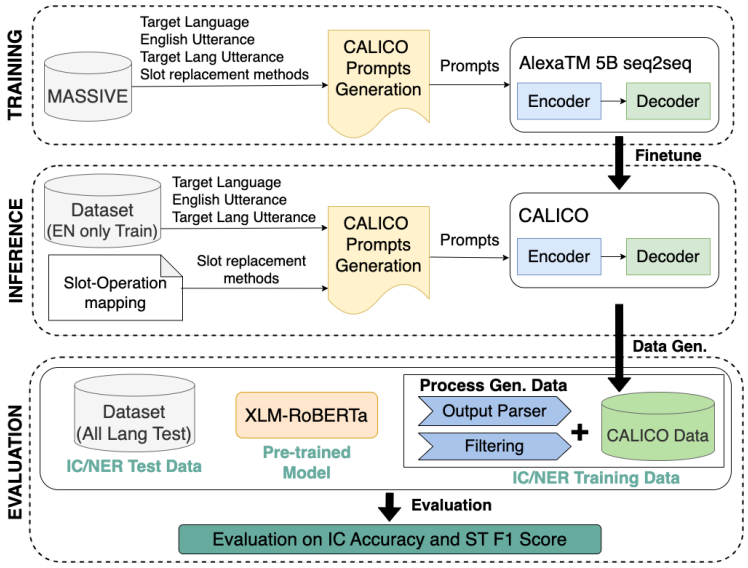

Figure 2: CALICO Training/Inference/Evaluation

2.3 Inference

For the target dataset, we take the English training data and instruct CALICO to generate corresponding data in the target language. We sample 8 outputs with top_p 0.95 and temperature 1.0 and then filter down to select only one as described next.

2.4 Iterative Filtering Mechanism (IFM)

We extend the post-processing pipeline of LINGUIST to include an Iterative Filtering Mechanism (IFM) to select higher quality samples from among the n-best CALICO-generated outputs.

Similar to LINGUIST, we first discard any outputs that do not pass heuristic string-based validation such as missing or extra brackets. We then apply English-IC filtering and backoff to the original English example in case no output is valid, in order to maintain the original per-intent distribution.

In the first round, we randomly select one of the remaining CALICO outputs and fine-tune the IC+ST task model on real English data plus the selected CALICO outputs in the other languages.

Then, we *re-select from among the* CALICO *n-best outputs*: we discard samples where the IC+ST model hypothesis disagrees with the intent and slot labels prompted for, then randomly select a single output again, using a different random seed to ensure that we don't always re-select the same output.

We apply this IFM repeatedly until performance plateaus (two iterations in all of our experiments).

3 Experiments

3.1 Models

We fine-tune AlexaTM 5B seq2seq model (Rosenbaum et al., 2022b; Soltan et al., 2022; FitzGerald et al., 2022a) as the CALICO data generaiton model. For the downstream task IC+ST model and iterative filtering model, we fine-tune `xlm-roberta-base` (Conneau et al., 2020) (12 layers, 768 hidden dimension), from the HuggingFace (Wolf et al., 2020) implementation.

3.2 Datasets

We fine-tune CALICO on cross-lingual prompts extracted from MASSIVE, which contains parallel IC+ST annotated data in 51 languages. We fine-tune on 6 languages: German, Spanish, French, Hindi, Japanese, and Portuguese, each parallel to English.

After the CALICO data generation model is trained, we apply it on two IC+ST datasets, MultiATIS++ and MultiSNIPS, to localize training data from English into the target languages. The CALICO model has never seen the specific intent and slot names, annotation scheme, or data conventions of the target downstream tasks, and therefore must generalize at inference.

3.2.1 MASSIVE

MASSIVE (FitzGerald et al., 2022b), Multilingual Amazon SLURP (SLU resource package) for Slot Filling, Intent Classification, and Virtual-Assistant Evaluation, contains 19,521 English realistic, labeled virtual-assistant utterances spanning 18 domains, 60 intents, and 55 slots. It is a parallel dataset, where each English utterance is localized or translated into 50 typologically diverse languages. *The dataset includes annotations on the human-chosen replacement method for each slot* (i.e. translation or localization or unchanged) for each pair of English and parallel target language utterance. Crucially, we use these slot-level annotations in the prompts for CALICO fine-tuning so that the model learns to follow instructions like `localization` (Figure 1).

Lang	**Eval on test data with HT slots**						
	Lower bound	**Upper bound**	**LINGUIST (our repro)**	**CALICO (No IFM)**	**CALICO (IFM)**	**CALICO (All Transl.)**	**IFM comb all**
	IC Accuracy (%)						
EN	97.10	97.54	97.77	97.66	97.77	**97.99**	97.88
DE	90.96	97.43	94.20	**97.10**	**97.10**	**97.10**	96.88
ES	95.09	96.76	96.76	97.32	**97.77**	97.43	97.54
FR	94.08	97.88	96.99	97.88	97.66	97.88	**98.21**
HI	85.38	94.64	90.62	92.41	**95.31**	92.30	94.75
JA	87.61	96.28	89.30	94.48	**96.85**	95.05	96.51
PT	91.63	96.88	96.21	95.98	**97.10**	96.32	96.65
AVG non-EN	90.79	96.65	94.01	95.86	**96.96**	96.01	96.76
	ST F1						
EN	95.73	96.01	95.69	95.71	**95.85**	95.76	95.68
DE	80.47	95.34	84.11	87.47	87.68	85.76	**89.23**
ES	81.53	87.55	84.43	86.81	**88.32**	87.26	87.57
FR	77.69	94.56	84.99	85.71	**86.59**	85.10	85.70
HI	64.45	87.81	77.88	76.92	76.32	82.69	**83.04**
JA	41.98	93.64	85.15	88.43	90.71	91.85	**92.38**
PT	50.50	92.16	**84.66**	81.90	83.83	81.06	83.12
AVG non-EN	66.10	91.84	83.54	84.54	85.58	85.62	**86.84**

Table 1: MultiATIS++ data generation performance on IC accuracy and ST F1 Score. The result is reported on Human-Translate ("HT") test set. The best (second best) result is bold (underlined).

Lang	**Eval on test data with HL slots**						
	Lower bound	**Upper bound**	**LINGUIST (our repro)**	**CALICO (No IFM)**	**CALICO (IFM)**	**CALICO (All Transl.)**	**IFM comb all**
	IC Accuracy (%)						
EN	97.10	97.54	97.77	97.66	97.77	**97.99**	97.88
DE	90.40	97.10	94.42	96.88	**97.21**	97.10	96.99
ES	95.54	96.43	96.88	96.88	96.76	96.76	**96.99**
FR	93.19	96.76	96.21	96.76	96.21	96.65	**96.88**
HI	86.16	93.08	91.29	92.86	94.98	93.53	**95.31**
JA	86.38	96.09	89.29	94.53	**95.98**	94.98	95.31
PT	91.52	96.09	95.54	95.20	95.76	95.54	**95.98**
AVG non-EN	90.53	95.93	93.94	95.52	96.15	95.76	**96.24**
	ST F1						
EN	95.73	96.01	95.69	95.71	**95.85**	95.76	95.68
DE	76.61	85.78	79.96	80.36	82.52	78.97	**83.54**
ES	71.86	76.08	74.11	83.57	**85.83**	78.14	84.50
FR	77.62	85.79	82.39	82.07	**83.10**	81.63	81.97
HI	72.84	84.93	76.52	78.62	80.79	79.99	**81.17**
JA	41.61	91.19	86.34	89.53	**91.61**	88.98	91.39
PT	77.65	90.18	83.63	82.69	**85.00**	81.25	84.07
AVG non-EN	69.70	85.66	80.49	82.81	**84.81**	81.49	84.44

Table 2: MultiATIS++ data generation performance on IC accuracy and ST F1 Score. The result is reported on new localized version ("HL") of the test set. The best (second best) result is bold (underlined).

Lang	Lower bound	Upper bound	LINGUIST	CALICO	CALICO (All Transl.)
	IC Accuracy (%)				
EN	99.01	98.86	98.86	99.01	98.79
ES	95.03	98.15	98.44	98.30	98.01
FR	96.59	98.58	98.72	98.44	97.94
HI	92.39	95.11	98.71	97.56	97.70
AVG non EN	94.67	97.28	98.62	98.10	97.88
	ST F1				
EN	93.63	95.48	95.24	95.35	95.11
ES	70.14	91.35	90.35	90.42	89.71
FR	70.47	86.73	84.35	88.81	88.53
HI	47.85	85.72	82.87	77.90	83.99
AVG non EN	62.82	87.93	85.86	85.71	87.41

Table 3: MultiSNIPS data generation performance on IC accuracy and ST F1 Score.

3.2.2 MultiATIS++

MultiATIS++ dataset extends the English-only Air Travel Information Services dataset ATIS to 9 languages via human translation. Our work focuses on the 7 languages MultiATIS++ shares with our pre-trained model: English (EN), German (DE), Spanish (ES), French (FR), Hindi (HI), Japanese (JA), Portuguese (PT), with 4488 (HI: 1440) training utterances, 490 (HI: 160) validation utterances, and 893 test utterances, over 18 (HI: 17) intents and 84 (HI: 75) slots. (The test set release also contains Turkish and Chinese.)

Since the original test set is human-translated from English, it contains only translated slot values (e.g. for "new orleans" in English, the Spanish version would be "nueva orleans"), but not *localized* entities. To showcase the effectiveness of CALICO, we create a **new version of the MultiATIS++ test set** in all 8 non-English languages, by asking human experts to *localize* the slot values for 8 slot types. Specifically, the human experts replace the original human-translated English slot value (e.g. "nueva orleans" in Spanish) with a value more appropriate value in the target language (e.g. "madrid"). The localized slot types are `airline_code`, `airline_name`, `airport_code`, `airport_name`, `city_name`, `country_name`, `state_code`, `state_name`.

We also ask the experts to modify the carrier phrase text as needed to make the entire request grammatically correct. For example, in French the experts might change the form of the definite article "le", "la", "les", or "l"' (all meaning "the") to match the plurality, gender, and pronunciation of the newly chosen slot value. The rest of the slot types and text remain as they are in the original.

3.2.3 MultiSNIPS

The MultiSNIPS dataset (Stickland et al., 2023) contains human translations of the SNIPS (Coucke et al., 2019) dataset into three languages: Spanish (ES), French (FR), and Hindi (HI).

3.3 Results

MultiATIS++ Results are shown in Tables 1 and 2, where the upper block reports IC Accuracy, and the lower block reports ST F1 Score. Table 1 is reported on the original test set, which contains human-translated slot values, whereas Table 2 is on our new version of the test set, where the slots are human-localized to versions more common in the target language.

"Lower bound" indicates the IC+ST model trained *only on the English training data*. "Upper bound" indicates the IC+ST model trained on MultiATIS++ training data for all 7 languages. The remaining columns indicate IC+ST models trained on the concatenation of original English training data with synthetic training data for the other 6 languages from one or more methods. "LINGUIST (our repro)" is our reproduction of LINGUIST.

"CALICO (IFM)" is our candidate approach, where we apply the `localization`, `translation`, and `copy` operations to specific slot values as shown in Table 6 (Appendix C), along with post-generation iterative filtering mechanism. "CALICO (All Transl.)" is our candidate model with the `translation` operation applied to all slots at inference time. In "CALICO (No IFM)", we do an ablation setup on the iterative filtering mechanism, where we do not perform iterative filtering. And finally in the column "IFM Comb all", we include the synthetic data from "CALICO (IFM)" combined with "LINGUIST (our repro)" and "CALICO (All Translate)".

We primarily focus on "AVG non EN", which is the average across the non-English languages. On the original test set (Table 1), we find that "CALICO (IFM)" is the best performing single method on average, surpassing LINGUIST by 2.95 points absolute on IC (from 94.01 to 96.96) and 2.04 points absolute on ST F1 (from 83.54 to 85.58).

The improvement of CALICO over LINGUIST is reflected on most languages, however Japanese (JA) shows the most improvement, having +7.55 / +5.56 (from 89.30 to 96.85 / from 85.15 to 90.71) points improvement on IC / ST. This suggests that Japanese was being limited the most by the drawbacks of LINGUIST, e.g. making translation mistakes out of context.

On the new localized test data (Table 2), both versions of CALICO improve over LINGUIST, and IFM improves even further. With CALICO plus IFM, we improve on IC from 93.94 LINGUIST to 96.15 (for 2.21 points absolute) and ST from 80.49 LINGUIST to 84.81 (i.e. 4.32 points absolute). As with the HT test set, the improvement is particularly large for Japanese (JA).

On both settings, we combine data from LINGUIST and both versions of CALICO, however find that the gains are not consistently synergistic.

We see the performance improvement of CALICO over LINGUIST is directly correlated with the "Success Rate" (Table 4 in Appendix B, filtering to keep only those outputs which pass string-matching heuristics and IC hypothesis filtering), indicating that the data produced from CALICO is cleaner and more usable than that of LINGUIST.

All data generation models and even the Upper Bound of including human-translated training data perform significantly worse on the test data with human-localized slots compared to the original human-translated test data, indicating that the human-localized test set is more challenging, and motivating future work on conversational agent localization.

MultiSNIPS Results are show in Table 3. Here, in the AVG non EN there are small differences overall compared to LINGUIST: CALICO (All Translate) is 0.74 points absolute worse on IC (from 98.62 to 97.88) and +1.55 points absolute better on ST (from 85.86 to 87.41). However, similarly to the MultiATIS++ results, CALICO (All Translate) out-performs LINGUIST. All models are close to the upper bound, however, indicating that this dataset may not be particularly challenging.

4 Conclusion and Future Work

We introduced CALICO, a novel pipeline for synthetic annotated data generation in new languages, via fine-tuning a largescale pre-trained multilingual seq2seq model. We demonstrated that unlike prior techniques that would translate slots out of context, CALICO can generate annotated slots based on the context and localize them with values more appropriate to the target language . In future, we plan to extend and leverage a reward model into a reinforcement learning setup to further improve the quality of the generated data. We would also like to explore ways to combine the positive effects of LINGUIST paraphrasing with CALICO localization.

References

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 597–604, Ann Arbor, Michigan. Association for Computational Linguistics.

Maximillian Chen, Alexandros Papangelis, Chenyang Tao, Seokhwan Kim, Andy Rosenbaum, Yang Liu, Zhou Yu, and Dilek Hakkani-Tur. 2023. PLACES: Prompting language models for social conversation synthesis. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 844–868, Dubrovnik, Croatia. Association for Computational Linguistics.

Maximillian Chen, Alexandros Papangelis, Chenyang Tao, Andy Rosenbaum, Seokhwan Kim, Yang Liu, Zhou Yu, and Dilek Hakkani-Tür. 2022. Weakly supervised data augmentation through prompting for dialogue understanding. In *NeurIPS 2022 Workshop on SyntheticData4ML.*

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling.

Eunah Cho, He Xie, and William M. Campbell. 2019. Paraphrase generation for semi-supervised learning in NLU. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neu-*

ral Language Generation, pages 45–54, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard 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, Online. Association for Computational Linguistics.

Alice Coucke, Mohammed Chlieh, Thibault Gisselbrecht, David Leroy, Mathieu Poumeyrol, and Thibaut Lavril. 2019. Efficient keyword spotting using dilated convolutions and gating.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3861–3867, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Jack FitzGerald, Shankar Ananthakrishnan, Konstantine Arkoudas, Davide Bernardi, Abhishek Bhagia, Claudio Delli Bovi, Jin Cao, RAKESH CHADA, Amit Chauhan, Luoxin Chen, Anurag Dwarakanath, Satyam Dwivedi, Turan Gojayev, Karthik Gopalakrishnan, Thomas Gueudre, Dilek Hakkani-Tur, Wael Hamza, Jonathan Hueser, Kevin Martin Jose, Haidar Khan, Beiye Liu, Jianhua Lu, Alessandro Manzotti, Pradeep Natarajan, Karolina Owczarzak, Gokmen Oz, Enrico Palumbo, Charith Peris, Chandana Satya Prakash, Stephen Rawls, Andy Rosenbaum, Anjali Shenoy, Saleh Soltan, Mukund Harakere, Liz Tan, Fabian Triefenbach, Pan Wei, Haiyang Yu, Shuai Zheng, Gokhan Tur, and Prem Natarajan. 2022a. Alexa teacher model: Pretraining and distilling multi-billion-parameter encoders for natural language understanding systems. In *KDD 2022*.

Jack FitzGerald, Christopher Leo Hench, Charith S. Peris, Scott Mackie, Kay Rottmann, A. Patricia Domínguez Sánchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, L. Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and P. Natarajan. 2022b. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *ArXiv*, abs/2204.08582.

Shailza Jolly, Tobias Falke, Caglar Tirkaz, and Daniil Sorokin. 2020. Data-efficient paraphrase generation to bootstrap intent classification and slot labeling for new features in task-oriented dialog systems. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 10–20, Online. International Committee on Computational Linguistics.

Manoj Kumar, Yuval Merhav, Haidar Khan, Rahul Gupta, Anna Rumshisky, and Wael Hamza. 2022. Controlled data generation via insertion operations for NLU. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 54–61, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26, Suzhou, China. Association for Computational Linguistics.

Nikolaos Malandrakis, Minmin Shen, Anuj Goyal, Shuyang Gao, Abhishek Sethi, and Angeliki Metallinou. 2019. Controlled text generation for data augmentation in intelligent artificial agents. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 90–98, Hong Kong. Association for Computational Linguistics.

Subhadarshi Panda, Caglar Tirkaz, Tobias Falke, and Patrick Lehnen. 2021. Multilingual Paraphrase Generation For Bootstrapping New Features in Task-Oriented Dialog Systems. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 30–39, Online.

Soham Parikh, Mitul Tiwari, Prashil Tumbade, and Quaizar Vohra. 2023. Exploring zero and few-shot techniques for intent classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 744–751, Toronto, Canada. Association for Computational Linguistics.

Andy Rosenbaum, Saleh Soltan, and Wael Hamza. 2023. Using large language models (llms) to synthesize training data. *Amazon Science*.

Andy Rosenbaum, Saleh Soltan, Wael Hamza, Marco Damonte, Isabel Groves, and Amir Saffari. 2022a. CLASP: Few-shot cross-lingual data augmentation for semantic parsing. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 444–462, Online only. Association for Computational Linguistics.

Andy Rosenbaum, Saleh Soltan, Wael Hamza, Yannick Versley, and Markus Boese. 2022b. LINGUIST: Language model instruction tuning to generate annotated utterances for intent classification and slot tagging. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 218–241, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Saleh Soltan, Shankar Ananthakrishnan, Jack G. M. FitzGerald, Rahul Gupta, Wael Hamza, Haidar Khan, Charith S. Peris, Stephen Rawls, Andrew Rosenbaum, Anna Rumshisky, Chandan Prakash, Mukund Sridhar, Fabian Triefenbach, Apurv Verma, Gokhan Tur, and Premkumar Natarajan. 2022. Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model. *ArXiv*, abs/2208.01448.

Asa Cooper Stickland, Sailik Sengupta, Jason Krone, Saab Mansour, and He He. 2023. Robustification of multilingual language models to real-world noise with robust contrastive pretraining. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Gokhan Tur and Renato De Mori. 2011. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. Wiley.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification 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 6382–6388, Hong Kong, China. 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.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268. Curran Associates, Inc.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual nlu.

Rongzhi Zhang, Yue Yu, and Chao Zhang. 2020. SeqMix: Augmenting active sequence labeling via sequence mixup. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8566–8579, Online. Association for Computational Linguistics.

Appendix

A Prompt Comparison

```
INPUT:
<language> Spanish </language>
<domain> travelinfo </domain>
<intent> flight </intent>
<include>
 [1 translation( pittsburgh ) ] ,
 [2 translation( new york ) ] ,
 [ 3 translation( 5 pm ) ]
</include>
<labels>
 [1=fromloc.city_name ] ,
 [2=toloc.city_name ] ,
 [3=depart_time.time
</labels>
<examples>
i need a flight
  from [1 pittsburgh ]
  to [2 new york ]
  leaving at [3 5 pm ]
</examples>

OUTPUTS:

1. necesito un vuelo
   desde [1 pittsburgh ]
   a [2 nueva york ]
   saliendo a las [3 cinco de la tarde ]
```

(a) The prompt with `translation` operation.

```
INPUT:
<language> Spanish </language>
<domain> travelinfo </domain>
<intent> flight </intent>
<include>
 [1 localization( pittsburgh ) ] ,
 [2 localization( new york ) ] ,
 [ 3 translation( 5 pm ) ]
</include>
<labels>
 [1=fromloc.city_name ] ,
 [2=toloc.city_name ] ,
 [3=depart_time.time
</labels>
<examples>
i need a flight
  from [1 pittsburgh ]
  to [2 new york ]
  leaving at [3 5 pm ]
</examples>

OUTPUTS:

1. necesito un vuelo
   desde [1 madrid ]
   a [2 barcelona ]
   saliendo a las [3 cinco de la tarde ]
```

(b) The prompt with `localization` operation.

Figure 3: For the same input, CALICO can follow the instruction prompt to map city names from English into Spanish either via literal translations (left) ("pittsburgh" → "pittsburgh" and "new york" → "nueva york") or via *localization* (right) ("pittsburgh" → "madrid" and "new york" → "barcelona").

B Success Rates

Table 4: MultiATIS++ data generation Success rate.

Method	Lang	Parse Success (%)	IC Filter Success (%)	Final Success (%)	# Utt. from Generation	# EN Utt. Copied	Total
LINGUIST (our repro)	DE	91.61	78.76	72.15	3037	1172	4209
	ES	91.59	80.54	73.77	3105	1104	4209
	FR	92.47	73.79	68.23	2872	1337	4209
	HI	89.69	79.00	70.85	2982	1227	4209
	JA	72.18	76.88	55.49	2336	1873	4209
	PT	92.33	83.30	76.91	3237	972	4209
	AVG	88.31	78.71	69.57	2928	1281	4209
CALICO	DE	99.44	90.78	90.27	4052	436	4488
	ES	99.44	94.25	93.72	4206	282	4488
	FR	99.84	94.14	93.99	4218	270	4488
	HI	97.82	85.16	83.30	3739	749	4488
	JA	95.10	79.48	75.58	3392	1096	4488
	PT	99.31	92.74	92.10	4133	355	4488
	AVG	98.49	89.43	88.16	3957	531	4488
CALICO (All Translate)	DE	99.35	91.35	90.76	4073	415	4488
	ES	98.93	94.01	93.00	4174	314	4488
	FR	99.64	94.43	94.09	4223	265	4488
	HI	97.08	85.76	83.26	3737	751	4488
	JA	94.45	82.66	78.07	3504	984	4488
	PT	98.95	92.96	91.99	4128	360	4488
	AVG	98.07	90.20	88.53	3973	515	4488

Table 5: MultiSNIPS data generation Success rate.

Method	Lang	Parse Success (%)	IC Filter Success (%)	Final Success (%)	# Utt. from Generation	# EN Utt. Copied	Total
LINGUIST (our repro)	ES	97.10	88.67	86.10	10846	1751	12597
	FR	97.30	97.74	95.10	11980	617	12597
	HI	72.86	92.34	67.28	8475	4122	12507
	AVG	89.09	92.92	82.83	10434	2163	12597
CALICO	ES	99.98	90.75	90.73	11278	1152	12430
	FR	99.98	97.61	97.59	12130	300	12430
	HI	99.73	92.66	92.41	11487	943	12430
	AVG	99.90	93.67	93.58	11632	798	12430
CALICO (All Translate)	ES	99.98	90.96	90.94	11304	1126	12430
	FR	99.99	97.45	97.44	12112	318	12430
	HI	99.85	92.30	92.16	11456	974	12430
	AVG	99.94	93.57	93.51	11624	806	12430

C CALICO Slot Operations

Table 6: mATIS++ CALICO Slot Operations

Slot	Operation	Slot	Operation
airline_code	copy	flight_mod	translation
airline_name	**localization**	flight_number	translation
airport_code	copy	flight_stop	translation
airport_name	**localization**	flight_time	translation
arrive_date.date_relative	translation	fromloc.airport_code	copy
arrive_date.day_name	translation	**fromloc.airport_name**	**localization**
arrive_date.day_number	translation	**fromloc.city_name**	**localization**
arrive_date.month_name	translation	fromloc.state_code	copy
arrive_date.today_relative	translation	**fromloc.state_name**	**localization**
arrive_time.end_time	translation	meal	translation
arrive_time.period_mod	translation	meal_code	copy
arrive_time.period_of_day	translation	meal_description	translation
arrive_time.start_time	translation	mod	translation
arrive_time.time	translation	month_name	translation
arrive_time.time_relative	translation	or	translation
booking_class	translation	period_of_day	translation
city_name	**localization**	restriction_code	copy
class_type	translation	return_date.date_relative	translation
compartment	translation	return_date.day_name	translation
connect	translation	return_date.day_number	translation
cost_relative	translation	return_date.month_name	translation
day_name	translation	return_date.today_relative	translation
day_number	translation	return_time.period_mod	translation
days_code	copy	return_time.period_of_day	translation
depart_date.date_relative	translation	round_trip	translation
depart_date.day_name	translation	state_code	copy
depart_date.day_number	translation	**state_name**	**localization**
depart_date.month_name	translation	stoploc.airport_code	copy
depart_date.today_relative	translation	**stoploc.airport_name**	**localization**
depart_date.year	translation	**stoploc.city_name**	**localization**
depart_time.end_time	translation	stoploc.state_code	copy
depart_time.period_mod	translation	time	translation
depart_time.period_of_day	translation	time_relative	translation
depart_time.start_time	translation	today_relative	translation
depart_time.time	translation	toloc.airport_code	copy
depart_time.time_relative	translation	**toloc.airport_name**	**localization**
economy	translation	**toloc.city_name**	**localization**
fare_amount	translation	**toloc.country_name**	**localization**
fare_basis_code	translation	toloc.state_code	copy
flight	translation	**toloc.state_name**	**localization**
flight_days	translation	transport_type	translation