Multilingual Paraphrase Generation For Bootstrapping New Features in Task-Oriented Dialog Systems

Subhadarshi Panda[1,*], **Caglar Tirkaz**[2], **Tobias Falke**[2] and **Patrick Lehnen**[2]
[1]Graduate Center, City University of New York, USA
[2]Amazon Alexa AI, Germany
spanda@gc.cuny.edu, {caglart,falket,plehnen}@amazon.com

Abstract

The lack of labeled training data for new features is a common problem in rapidly changing real-world dialog systems. As a solution, we propose a multilingual paraphrase generation model that can be used to generate novel utterances for a target feature and target language. The generated utterances can be used to augment existing training data to improve intent classification and slot labeling models. We evaluate the quality of generated utterances using intrinsic evaluation metrics and by conducting downstream evaluation experiments with English as the source language and nine different target languages. Our method shows promise across languages, even in a zero-shot setting where no seed data is available.

1 Introduction

Spoken language understanding is a core problem in task oriented dialog systems with the goal of understanding and formalizing the intent expressed by an utterance (Tur and De Mori, 2011). It is often modeled as intent classification (IC), an utterance-level multi-class classification problem, and slot labeling (SL), a sequence labeling problem over the utterance's tokens. In recent years, approaches that train joint models for both tasks and that leverage powerful pre-trained neural models greatly improved the state-of-the-art performance on available benchmarks for IC and SL (Louvan and Magnini, 2020; Weld et al., 2021).

A common challenge in real-world systems is the problem of *feature bootstrapping*: If a new feature should be supported, the label space needs to be extended with new intent or slot labels, and the model needs to be retrained to learn to classify corresponding utterances. However, labeled examples for the new feature are typically limited to a small set of seed examples, as the collection of more annotations would make feature expansion costly and slow. As a possible solution, previous work explored the automatic generation of paraphrases to augment the seed data (Malandrakis et al., 2019; Cho et al., 2019; Jolly et al., 2020).

In this work, we study feature bootstrapping in the case of a *multilingual dialog system*. Many large-scale real-world dialog systems, e.g. Apple's Siri, Amazon's Alexa and Google's Assistant, support interactions in multiple languages. In such systems, the coverage of languages and the range of features is continuously expanded. That can lead to differences in the supported intent and slot labels across languages, in particular if a new language is added later or if new features are not rolled out to all languages simultaneously. As a consequence, labeled data for a feature can be available in one language, but limited or completely absent in another. With multilingual paraphrase generation, we can benefit from this setup and improve data augmentation for data-scarce languages via cross-lingual transfer from data-rich languages. As a result, the data augmentation can not only be applied with seed data, i.e. in a *few-shot* setting, but even under *zero-shot* conditions with no seeds at all for the target language.

To address this setup, we follow the recent work of Jolly et al. (2020), which proposes to use an encoder-decoder model that maps from structured meaning representations to corresponding utterances. Because such an input is language-agnostic, it is particularly well-suited for the multilingual setup. We make the following extensions: First, we port their model to a transformer-based architecture and allow multilingual training by adding the desired target language as a new input to the conditional generation. Second, we let the model generate slot labels along with tokens to alleviate the need for additional slot projection techniques. And third, we introduce improved paraphrase decoding methods that leverage a model-based selec-

[*] The first author worked on this paper during an internship at Amazon Alexa AI.

tion strategy. With that, we are able to generate labeled data for a new feature even in the zero-shot setting where no seeds are available at all.

We evaluate our approach by simulating a cross-lingual feature bootstrapping setting, either few-shot or zero-shot, on MultiATIS, a common IC/SL benchmark spanning nine languages. The experiments compare against several alternative methods, including previous work for mono-lingual paraphrase generation and machine translation. We find that our method produces paraphrases of high novelty and diversity and using it for IC/SL training shows promising downstream classification performance.

2 Related work

Various studies have explored paraphrase generation for dialog systems. Bowman et al. (2016) showed that generating sentences from a continuous latent space is possible using a variational autoencoder model and provided guidelines on how to train such a generation model. However, our model uses an encoder-decoder approach which can handle the intent and language as categorical inputs in addition to the sequence input. Malandrakis et al. (2019) explored a variety of controlled paraphrase generation approaches for data augmentation and proposed to use conditional variational autoencoders which they showed obtained the best results. Our method is different as it uses a conditional seq2seq model that can generate text from any sequence of slots and does not require an utterance as an input. Xia et al. (2020) propose a transformer-based conditional variational autoencoder for few shot utterance generation where the latent space represents the intent as two independent parts (domain and action). Our approach is different since it models the language and intent of the generation that can be controlled explicitly. Also, our model is the first to enable zero-shot utterance generation. Cho et al. (2019) generate paraphrases for seed examples with a transformer seq2seq model and self-label them with a baseline intent and slot model. We follow a similar approach but our model generates utterances from a sequence of slots rather than an utterance, which enables an explicitly controlled generation. Also the number of seed utterances we use is merely 20 for the few shot setup unlike around 1M seed para-carrier phrase pairs in Cho et al. (2019).

Several other studies follow a text-to-text approach and assume training data in the form of paraphrase pairs for training paraphrase generation models in a single language (Gupta et al., 2018; Li et al., 2018, 2019). Our approach is focused towards generating utterances in the dialog domain that can generate utterances from a sequence of slots conditioned on both intent and language.

Jolly et al. (2020) showed that an interpretation-to-text model can be used with shuffling-based sampling techniques to generate diverse and novel paraphrases from small amounts of seed data, that improve accuracy when augmenting to the existing training data. Our approach is different as our model can generate the slot annotations along with the the utterance, which are necessary for the slot labeling task. Our model can be seen as an extension of the model by Jolly et al. (2020) to a transformer based model, with the added functionality of controlling the language in which the utterance generation is needed, which in turn enables zero shot generation.

Using large pre-trained models has also been shown to be effective for paraphrase generation. Chen et al. (2020) for instance show the effectiveness of using GPT-2 (Radford et al., 2019) for generating text from tabular data (a set of attribute-value pairs). Our model, however, does not rely on pre-trained weights from another model such as GPT-2, is scalable, and can be applied to training data from any domain, for instance, dialog domain.

Beyond paraphrase generation, several other techniques have been proposed for feature bootstrapping. Machine translation can be used from data-rich to data-scarce languages (Gaspers et al., 2018; Xu et al., 2020). Cross-lingual transfer learning can also leverage use existing data in other languages (Do and Gaspers, 2019). If a feature is already being actively used, feedback signals from users, such as paraphrases or interruptions, can be used to obtain additional training data (Muralidharan et al., 2019; Falke et al., 2020).

3 Proposed method

We want to augment existing labeled utterances by generating additional novel utterances in a desired target language. In our case, existing data consists of feature-unrelated data (intents and slots already supported) spanning all languages and feature-related data, which is available in a source language but is small (few-shot) or not available (zero shot) in other languages. For generation, we first extract

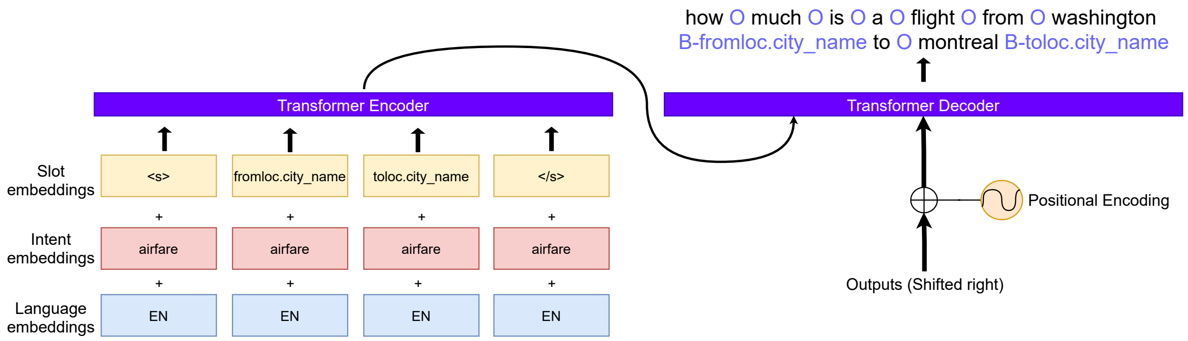


Figure 1: **Overall architecture of the multilingual paraphrase generation model.** The slot, intent and language embeddings are added at the slot level to obtain representations to input to the encoder. The <s> and </s> tags are necessary as they enable handling cases where we want to generate paraphrases having no associated slots. The decoder generates the slot labels along with the paraphrase tokens.

the intent and slot types from the available data. We then generate a new utterance by conditioning a multilingual language model on the intent, slot types and the target language. We refer to utterances that have the same intent and slot types as *paraphrases* of each other since they convey the same meaning in the context of the SLU system.

3.1 Paraphrase Generation Model

In order to generate paraphrases, we train a multilingual paraphrase generation model that generates a paraphrase given a language, an intent and a set of slot types. The model architecture is outlined in Figure 1. The model uses self-attention based encoder and decoder similar to the transformer (Vaswani et al., 2017). The encoder of the model receives as input the language embedding and the intent embedding, which are added to the slot embedding. Unlike the transformer model (Vaswani et al., 2017), we do not use the positional embedding in the encoder. This is because the order of the slot types in the input sequence does not matter and is thus made indistinguishable for the encoder.

In order to generate paraphrases which can be used for data augmentation, we would need the slot annotations and the intents of the generations. Note that we already know the intent of the generated paraphrase since it is the same intent as specified while generating it. The slot annotations, however, are not readily obtained from the input slot types. We can make the slot annotations part of the output sequence by generating the slot label in BIO format in every alternate time step, which would be the slot label for the token generated in the previous time step. This enables the model to generate the slot annotations along with the paraphrase. An illustrative example is shown in Figure 1.

3.2 Decoding Techniques

Generating the output sequence token-by-token can be done by using greedy decoding where given learned model parameters θ, the most likely token is picked at each decoding step as $x_t = \text{argmax}\ p_\theta(x_t|x_{<t})$. Such a generation process is deterministic. For our task of generating paraphrases, we are interested in generating diverse and novel utterances. Non-deterministic sampling methods such as top-k sampling has been used in related work (Fan et al., 2018; Welleck et al., 2020; Jolly et al., 2020) to achieve this. In top-k random sampling, we first scale the logits z_w by using a temperature parameter τ before applying softmax.

$$p(x_t = w|x_{<t}) = \frac{\exp(z_w/\tau)}{\sum_{w' \in V} \exp(z_{w'}/\tau)}, \qquad (1)$$

where V is the decoder's vocabulary. Setting $\tau > 1$ encourages the resulting probability distribution to be less spiky, thereby encouraging diverse choices during sampling. The top-k sampling restricts the size of the most likely candidate pool to $k \leq |V|$.

3.3 Balanced Augmentation

The generated paraphrases can be used to augment the existing training data. Since the training data we use is highly imbalanced, data augmentation might lead to disturbance in the original intent distribution. To ensure that the data augmentation process does not disturb the original intent distribution, we compute the number of samples to augment using the following constraint: the ratio of target intent to other intents for the target language should be the same as the ratio of target intent to other intents in the source language. Sometimes, using the above constraint results in a negligible number of samples for augmentation, in which cases we use a minimal number of samples (see experiments).

3.4 Paraphrase Selection

In addition to deciding how many paraphrases to augment, it is also crucial to decide which paraphrases to use. Preliminary experimental results showed that samping uniformly from all generated paraphrases does not lead to improvement over the baseline. Upon manual examination we found that not all the paraphrases belong to the desired target intent. To cope with that problem, we use the baseline downstream intent classification and slot labeling model, which is trained only on the existing data, to compute the likelihood of the generated paraphrases to belong to the target intent. We rank all the generated paraphrases based on these probabilities and select from the top of the pool for augmentation of the seed data.

4 Experimental setup

We evaluate our approach by simulating few-shot and zero-shot feature bootstrapping scenarios.

4.1 Data

We use the MultiATIS++ data (Xu et al., 2020), a parallel IC/SL corpus that was created by translating the original English dataset. It covers a total of 9 languages: English, Hindi, Turkish, German, French, Portuguese, Spanish, Japanese and Chinese. The languages encompass a diverse set of language families: Indo-European, Sino-Tibetan, Japonic and Altaic.

Choosing target intents To reduce the number of experiments, we only choose three different intents for simulating the feature bootstrapping scenario. The MultiATIS++ dataset is highly imbalanced in terms of intent frequencies. For instance, 74% of the English training data has the intent *atis_flight* and as many as 9 intents have less than 20 training samples. The trend is similar for the non-English languages. For choosing target intents for simulating the zero shot and few shot training data, we therefore consider the following three target intents: (a) *atis_airfare*, which is highly frequent, (b) *atis_airline*, which has medium frequency, and (c) *atis_city* which is scarce.

Preprocessing We remove the samples in the MultiATIS++ data for which the number of tokens and the number of slot values do not match.[1] We also only consider the first intent for the samples that have multiple intent annotations. We show the data sizes after preprocessing in Table 1.

Training setup To simulate the feature bootstrapping scenario, we consider only 20 samples (**few shot setup**) or no samples at all (**zero shot setup**) from the MultiATIS++ data for a specific target intent in a target language.[2]

Language setup We use English as the source language and consider 8 target languages (Hindi, Turkish, German, French, Portuguese, Spanish, Japanese, Chinese) simultaneously. This encourages the model parameters to be shared across all the 9 languages including the source language English. The purpose of this setup is to enable us to study the knowledge transfer across multiple target languages in addition to that from the source language. We train a single model for paraphrase generation on all the languages as well as a single multi-lingual downstream IC/SL model.

4.2 Models and Training Details

Paraphrase generation training Since the training data is imbalanced, we balanced the training data by oversampling the intents to match the frequency of the most frequent intent.[3] For both the encoder and the decoder, the multi-head attention layers' hidden dimension was set to 128 and the position-wise feed forward layers' hidden dimension was set to 256. The number of encoder and decoder layers was set to 3 each. The number of heads was set to 8. Dropout of 0.1 was used in both the encoder and the decoder. The model parameters were initialized with Xavier initialization (Glorot and Bengio, 2010). The model was trained using Adam optimizer (Kingma and Ba, 2014) with a learning rate of 5e-4 and a gradient clipping of 1. The training was stopped when the development loss did not improve for 5 epochs.

Generating paraphrases For generating paraphrases in the target intent in the target language,

[1] This leads to removal of 0.6% of the total samples.

[2] For cases that have less than 20 samples to pick from, we consider all the samples which are available.

[3] Experiments with the original imbalanced training data resulted in generating paraphrases which belongs to one of the frequent intents, even if the desired intent was one with a low frequency in the training data.

Language	Train	Dev	Test	Unique intents (Train)	Unique slots (Train)
DE	4,487	490	892	17	79
EN	4,488	490	893	17	79
ES	4,484	489	813	17	79
FR	4,413	489	791	17	79
HI	1,495	160	893	16	70
JA	4,487	490	886	17	78
PT	4,478	489	892	17	79
TR	626	60	715	15	62
ZH	4,487	490	893	17	79

Table 1: MultiATIS++ data statistics.

we used the slots appearing in the existing training data in the target intent. We used greedy decoding and top-k sampling with $k = 3, 5, 10$ and $\tau = 1.0, 2.0$. For a given input, we generated using the top-k random sampling three times with different random seeds. We finally combined all generations and ranked the candidates using the baseline downstream system's prediction probability. The number of paraphrases that are selected is determined as in 3.3, with 20 as the minimum.

Methods for comparison We compare our method against four alternatives:
(a) Baseline: No data augmentation at all. The downstream model is trained using just the available seed examples for the target intent.
(b) Oversampling: We oversample the samples per intent uniformly at random to match the size of the augmented training data using the proposed method. This is only applicable to the few shot setup since for the zero shot setup, there are no existing samples in the target intent in the target language to sample from.
(c) CVAE seq2seq model: We generate paraphrases using the CVAE seq2seq model by Malandrakis et al. (2019). The original CVAE seq2seq model as proposed by Malandrakis et al. (2019) defines the set {domain, intent, slots} as the *signature* of an utterance and denotes the carrier phrases for a given signature to be paraphrases. These carrier phrases are then used to create input-output pairs for the CVAE seq2seq model training. Since the original formulation does not take into account the language of generation, we adapt the method for our case by defining the signature as the set {language, intent, slots}. We set the model's hidden dimension to 128, used the 100-dimensional GloVe embeddings (Pennington et al., 2014) pretrained on Wikipedia, and trained the model without freezing embeddings using early stopping with a patience of 5 epochs by monitoring the development loss. Finally we generated 100 carrier phrases for each carrier phrase input in the target intent in the target language. Paraphrases were obtained by injecting the slot values to the generated carrier phrases. The pool of all paraphrases was sorted using the baseline downstream system's prediction probabilities. The CVAE seq2seq model was only applicable to the few shot setup since in the zero shot setup there are no existing carrier phrases in the target language in the target intent that can be used to sample from.
(d) Machine translation: We augmented the translations generated from English using the MT+fast-align approach from the MultiATIS++ paper (Xu et al., 2020). For the few shot setup, we added all the translated utterances except the ones that correspond to those utterances we already picked as the few shot samples. For the zero shot setup, we added all the translated utterances.

Downstream training Unlike the paraphrase generation model training, we do not balance the simulated training data by oversampling based on intent. This choice was made to make sure that the original intent distribution was preserved for the downstream model training. We used the BERT base multilingual cased model (Devlin et al., 2019)[4] and added an intent head and a slot head on top for joint intent classification and slot labeling. Each head uses a hidden size of 256 and ReLU activation. The model was trained using Adam optimizer with a learning rate of 0.1. The training was stopped when the development semantic error rate (Su et al., 2018) did not improve for 3 epochs.

4.3 Intrinsic evaluation metrics

We evaluate the quality of the generated paraphrases using the following metrics. Let S be the set of input slot types and G be the set of generated slot types.

[4] https://github.com/google-research/bert/blob/master/multilingual.md

Few shot results:							
Generation method	**All retrieval score**	**Exact match**	**Partial match**	**F1 slot score**	**Jaccard index**	**Novelty**	**Diversity**
greedy	0.58	0.43	0.94	0.76	0.68	0.96	0
τ = 1.0, topk = 3	0.54	0.4	0.9	0.71	0.64	0.98	0.92
τ = 1.0, topk = 5	0.52	0.37	0.87	0.69	0.61	0.98	0.94
τ = 1.0, topk = 10	0.52	0.38	0.87	0.69	0.61	0.98	0.96
τ = 2.0, topk = 3	0.49	0.3	0.86	0.64	0.55	0.99	0.98
τ = 2.0, topk = 5	0.44	0.23	0.86	0.59	0.49	0.99	1
τ = 2.0, topk = 10	0.44	0.23	0.84	0.57	0.48	1	1
Zero shot results:							
Generation method	**All retrieval score**	**Exact match**	**Partial match**	**F1 slot score**	**Jaccard index**	**Novelty**	**Diversity**
greedy	0.41	0.22	0.78	0.56	0.48	1	0
τ = 1.0, topk = 3	0.4	0.26	0.78	0.58	0.5	1	0.95
τ = 1.0, topk = 5	0.4	0.22	0.78	0.56	0.48	1	0.96
τ = 1.0, topk = 10	0.39	0.19	0.77	0.53	0.44	1	0.98
τ = 2.0, topk = 3	0.38	0.16	0.76	0.5	0.41	1	0.99
τ = 2.0, topk = 5	0.35	0.1	0.73	0.45	0.35	1	1
τ = 2.0, topk = 10	0.34	0.12	0.72	0.43	0.34	1	1

Table 2: Intrinsic evaluation scores for different generation methods in few shot and zero shot scenarios.

	Few shot						**Zero shot**		
Language	**Lang. detection score**		**Novelty**		**Diversity**		**Lang. detection score**	**Novelty**	**Diversity**
	CVAE	Ours	CVAE	Ours	CVAE	Ours	Ours		
DE	0.69	0.95	0.43	0.97	0.33	0.81	0.97	1	0.85
ES	0.71	0.91	0.48	0.98	0.44	0.82	0.93	1	0.83
FR	0.78	0.94	0.47	0.98	0.36	0.82	0.95	1	0.85
HI	0.69	0.97	0.5	0.97	0.28	0.81	0.97	1	0.81
JA	0.83	0.96	0.52	1	0.39	0.85	1	1	0.85
PT	0.5	0.75	0.55	0.97	0.38	0.81	0.86	1	0.85
TR	0.01	0.34	0.25	0.99	0.22	0.85	0.53	1	0.84
ZH	0.57	0.68	0.61	1	0.52	0.85	0.62	1	0.85

Table 3: Intrinsic evaluation scores for different target languages in few shot and zero shot scenarios.

All retrieval score The all retrieval score r measures if all the input slots were retrieved in the generation.

$$r = \begin{cases} 1 & \text{if } |S \cap G| = |S| \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Exact match The exact match score r measures if all the input slots and output slots exactly match (Malandrakis et al., 2019).

$$r = \begin{cases} 1 & \text{if } S = G \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Partial match The partial match score r measures if at least one output slot matches an input slot.

$$r = \begin{cases} 1 & \text{if } |S \cap G| > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

F1 slot score The F1 slot score F_1 measures the set similarity between S and G using precision and recall which are defined for sets as follows.

$$\text{precision} = \frac{|S \cap G|}{|G|}, \text{recall} = \frac{|S \cap G|}{|S|} \tag{5}$$

Jaccard index Jaccard index measures the set similarity between S and G as their intersection size divided by the union size.

Novelty Let P be the set of paraphrases generated from a base utterance u.

$$\text{novelty} = \frac{1}{|P|} \sum_{u' \in P} \Big(1 - \text{BLEU4}(u, u')\Big) \tag{6}$$

Diversity The diversity is computed using the generated paraphrases P.

$$\text{diversity} = \frac{\sum_{u' \in P, u'' \in P, u' \neq u''} \Big(1 - \text{BLEU4}(u', u'')\Big)}{|P| \times (|P| - 1)} \tag{7}$$

Language detection score We are interested in quantifying if a generated paraphrase is in the target language. We use `langdetect`[5] to compute $p(\text{lang} = \text{target lang})$. Higher scores denote better language generation.

[5] https://github.com/Mimino666/langdetect

	Method	**Intent classification**								
		DE	ES	FR	HI	JA	PT	TR	ZH	AVG.
Few shot	Baseline	95.3	79.2	**86.0**	72.9	69.6	80.4	66.3	56.7	75.8
	Oversampling	93.6	80.6	84.4	78.2	**76.9**	83.7	63.1	62.3	77.8
	CVAE	94.1	84.7	83.7	71.7	63.8	**85.6**	**69.1**	55.8	76.1
	MT	91.3	**86.7**	81.6	**83.4**	70.9	82.8	64.0	56.0	77.1
	Paraphrasing	**97.1**	79.5	84.4	75.9	74.1	80.8	65.5	**66.8**	**78.0**
Zero shot	Baseline	48.7	44.8	51.6	15.0	3.1	33.8	8.0	1.0	25.8
	MT	**75.6**	**87.8**	**77.9**	**86.4**	**8.3**	**82.8**	**57.6**	**23.8**	**62.5**
	Paraphrasing	56.9	50.5	51.6	34.3	7.4	26.9	12.3	4.0	30.5

Table 4: Downstream intent classification accuracies (%). Each score shown is the average score of 10 runs.

	Method	**Slot labeling**								
		DE	ES	FR	HI	JA	PT	TR	ZH	AVG.
Few shot	Baseline	**98.0**	**85.0**	91.3	74.6	89.6	**92.2**	79.6	90.6	**87.6**
	Oversampling	96.2	84.6	**91.7**	**76.9**	89.9	90.0	81.3	90.3	**87.6**
	CVAE	97.3	83.5	90.0	75.8	**90.8**	91.6	**82.1**	89.6	**87.6**
	MT	95.0	78.8	90.9	73.0	90.8	82.9	77.9	88.8	84.8
	Paraphrasing	97.2	80.8	89.7	76.2	90.2	91.3	78.6	**91.6**	86.9
Zero shot	Baseline	**93.9**	**84.5**	89.3	72.5	89.1	88.7	77.3	87.8	85.4
	MT	92.0	79.9	88.5	**73.3**	**92.1**	82.1	76.2	**88.7**	84.1
	Paraphrasing	93.1	84.1	**90.5**	70.3	89.5	**91.5**	**77.6**	87.2	**85.5**

Table 5: Downstream slot labeling F1 scores (%). Each score shown is the average score of 10 runs.

5 Experimental results

5.1 Intrinsic Evaluation

For both the few shot and zero shot setups, the paraphrases used for intrinsic evaluation are generated in the target intent and the target language only. For the top-k sampling based generation, we generate for each input three times with different random seeds and compute novelty and diversity scores.

Table 2 shows intrinsic evaluation results for different generation methods. For the few shot setup, the all retrieval, exact match, partial match, F1 slot and Jaccard index scores decrease upon increasing top-k and temperature. The highest scores for the above metrics are obtained for the greedy generation, which indicates that the generated slot types are most similar to the input slot types in that case. However, it is the opposite for the novelty and diversity metrics where the scores are higher with larger top-k and temperatures. For the zero shot setup, the overall trend is similar to the few shot setup. The slot similarity based metrics are lower in general, which indicates that even as little as 20 samples in the few shot setup improve the generation of desired slots. The novelty scores for the zero shot setup are 1 as we would expect.

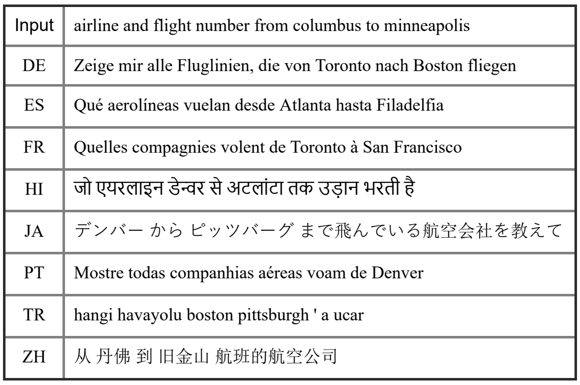
In Table 3, we show that the intrinsic evaluation results using the proposed approach are consistently better than the CVAE seq2seq paraphrase generation model (Malandrakis et al., 2019). The language detection score varies across languages, which may be due to the vocabulary overlap between languages, e.g., *San Francisco* appears in both English and German utterances. Interestingly we also observe code switching, i.e. mixed-language generations, while using our approach.

5.2 Downstream Evaluation

We evaluate the downstream intent classification using accuracy and the slot labeling using F1 score. Since we are interested in measuring the variation in scores for the target intents, we only report the scores for the test samples in the target intents in Tables 4 and 5. We run each downstream training experiment 10 times and report the mean scores for each language and also the average across languages in the AVG column in Tables 4 and 5. We are also interested in tracking the scores for the test samples having intents other than the target intents since we need to ensure that the scores on the other intents does not go down. We found that the effect on the scores (both intent classification and slot labeling) for the other intents is negligible using paraphrasing and other methods.[6]

In Tables 4 and 5, our paraphrasing results outperform the baseline scores on average. In the few shot setup, our paraphrasing approach outperforms the CVAE seq2seq approach in 6 (DE, ES, FR, HI, JA, ZH) out of 8 languages in intent classification and overall obtains an improvement of 1.9% intent classification accuracy across all target languages.

[6] The maximum drop in score was less than 1% absolute.



Input	airline and flight number from columbus to minneapolis
DE	Zeige mir alle Fluglinien, die von Toronto nach Boston fliegen
ES	Qué aerolíneas vuelan desde Atlanta hasta Filadelfia
FR	Quelles compagnies volent de Toronto à San Francisco
HI	जो एयरलाइन डेन्वर से अटलांटा तक उड़ान भरती है
JA	デンバー から ピッツバーグ まで飛んでいる航空会社を教えて
PT	Mostre todas companhias aéreas voam de Denver
TR	hangi havayolu boston pittsburgh ' a ucar
ZH	从 丹佛 到 旧金山 航班的航空公司

Table 6: Examples of paraphrases generated using the multilingual paraphrase generation model for *airline* and slots *fromloc* and *toloc*. The paraphrases shown are cherry picked from a set of generations.

Both oversampling and MT approaches are competitive. Oversampling performs the best for JA whereas MT performs the best for ES and HI. Our paraphrasing approach results in the best intent classification scores overall (78%). In terms of slot F1 scores, we see mixed results with no clear best method (baseline, oversampling and CVAE all result in 87.6% F1 score). Notably, the MT approach results in the lowest overall slot F1 score of just 84.8% on average.

In the zero shot setup, the MT approach outperforms our paraphrasing approach by a large margin in intent classification (62.5%). However we note that the paraphrasing approach requires no dependencies on other models or other data, unlike the MT approach which requires a parallel corpus to train the MT model. In terms of slot F1 scores, our paraphrasing approach and the baseline approach both result in almost similar overall scores (85.5% and 85.4%), both higher than the MT approach. The lower slot F1 scores using the MT approach in few and zero shot setups indicate that the fast align method to align slots in source and translation might result in noisy training data affecting the SL model.

5.3 Examples

Paraphrases generated in different languages for a given input are shown in Table 6. The intent is *airline* and the slots are *fromloc.city_name* for *columbus* and *toloc.city_name* for *minneapolis*. For this intent and the slots, the generated paraphrase in German (translated to English) is *Show me all the airlines that fly from Toronto to Boston*. The desired intent, that is *airline* is realized in the generated paraphrase. Additionally, *Toronto* and *Boston* are the slot values respectively for the slot types *fromloc.city_name* and *toloc.city_name*. For Spanish, the generated paraphrase (translated to English) is *Which Airlines Fly from Atlanta to Philadelphia*. The *airline* intent is realized in the generated paraphrase and also *Atlanta* and *Philadelphia* are the slot values produced associated with the desired slot types. As illustrated by the examples, the model is free to pick a specific slot value during generation, leading to variations across languages, but all are consistent with the slot type.

6 Conclusion

In this paper, we proposed a multilingual paraphrase generation model that can be used for feature bootstrapping with or without seed data in the target language. In addition to generating a paraphrase, the model also generates the associated slot labels, enabling the generation to be used directly for data augmentation to existing training data. Our method is language agnostic and scalable, with no dependencies on pre-trained models or additional data. We validate our method using experiments on the MultiATIS++ dataset containing utterances spanning 9 languages. Intrinsic evaluation shows that paraphrases generated using our approach have higher novelty and diversity in comparison to CVAE seq2seq based paraphrase generation. Additionally, downstream evaluation shows that using the generated paraphrases for data augmentation results in improvements over baseline and related techniques in a wide range of languages and setups. To the best of our knowledge, this is the first successful exploration of generating paraphrases for SLU in a cross-lingual setup.

In the future, we would like to explore strategies to exploit monolingual data in the target languages to further refine the paraphrase generation. We would also like to leverage pre-trained multilingual text-to-text models such as mT5 (Xue et al., 2020) for multilingual paraphrase generation in the dialog system domain.

Acknowledgements

We would like to thank our anonymous reviewers for their thoughtful comments and suggestions that improved the final version of this paper.

References

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020. Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Online. Association for Computational Linguistics.

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

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

Quynh Do and Judith Gaspers. 2019. Cross-lingual transfer learning with data selection for large-scale spoken language understanding. 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 1455–1460, Hong Kong, China. Association for Computational Linguistics.

Tobias Falke, Markus Boese, Daniil Sorokin, Caglar Tirkaz, and Patrick Lehnen. 2020. Leveraging user paraphrasing behavior in dialog systems to automatically collect annotations for long-tail utterances. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 21–32, Online. International Committee on Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Judith Gaspers, Penny Karanasou, and Rajen Chatterjee. 2018. Selecting machine-translated data for quick bootstrapping of a natural language understanding system. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 137–144, New Orleans - Louisiana. Association for Computational Linguistics.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

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.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878, Brussels, Belgium. Association for Computational Linguistics.

Zichao Li, Xin Jiang, Lifeng Shang, and Qun Liu. 2019. Decomposable neural paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3403–3414, Florence, Italy. Association for Computational Linguistics.

Samuel Louvan and Bernardo Magnini. 2020. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 480–496, Barcelona, Spain (Online). International Committee on 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.

Deepak Muralidharan, Justine Kao, Xiao Yang, Lin Li, Lavanya Viswanathan, Mubarak Seyed Ibrahim, Kevin Luikens, Stephen Pulman, Ashish Garg, Atish

Kothari, and Jason Williams. 2019. Leveraging User Engagement Signals For Entity Labeling in a Virtual Assistant. *arXiv*, 1909.09143.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Chengwei Su, Rahul Gupta, Shankar Ananthakrishnan, and Spyros Matsoukas. 2018. A re-ranker scheme for integrating large scale nlu models. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 670–676.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

H. Weld, X. Huang, S. Long, J. Poon, and S. C. Han. 2021. A survey of joint intent detection and slot-filling models in natural language understanding. *arXiv*, 2101.08091.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Congying Xia, Caiming Xiong, Philip Yu, and Richard Socher. 2020. Composed variational natural language generation for few-shot intents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3379–3388, Online. Association for Computational Linguistics.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual NLU. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5052–5063, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mT5: A massively multilingual pre-trained text-to-text transformer.