Multi-level Contrastive Learning for Cross-lingual Spoken Language Understanding

Shining Liang\textsuperscript{1,2,*}, Linjun Shou\textsuperscript{3}, Jian Pei\textsuperscript{4}, Ming Gong\textsuperscript{3}, Wanli Zuo\textsuperscript{1,2}, Xianglin Zuo\textsuperscript{1,2†}, Daxin Jiang\textsuperscript{3†}

\textsuperscript{1}College of Computer Science and Technology, Jilin University
\textsuperscript{2}Key laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education
\textsuperscript{3}STCA NLP Group, Microsoft
\textsuperscript{4}School of Computing Science, Simon Fraser University
\{liangsn17,zuoxl17\}@mails.jlu.edu.cn, \{migon,lisho,djiang\}@microsoft.com, zuowl@jlu.edu.cn

Abstract

Although spoken language understanding (SLU) has achieved great success in high-resource languages, such as English, it remains challenging in low-resource languages mainly due to the lack of high quality training data. The recent multilingual code-switching approach samples some words in an input utterance and replaces them by expressions in some other languages of the same meaning. The multilingual code-switching approach achieves better alignments of representations across languages in zero-shot cross-lingual SLU. Surprisingly, all existing multilingual code-switching methods disregard the inherent semantic structure in SLU, i.e., most utterances contain one or more slots, and each slot consists of one or more words. In this paper, we propose to exploit the “utterance-slot-word” structure of SLU and systematically model this structure by a multi-level contrastive learning framework at the utterance, slot, and word levels. We develop novel code-switching schemes to generate hard negative examples for contrastive learning at all levels. Furthermore, we develop a label-aware joint model to leverage label semantics for cross-lingual knowledge transfer. Our experimental results show that our proposed methods significantly improve the performance compared with the strong baselines on two zero-shot cross-lingual SLU benchmark datasets.

1 Introduction

Aiming at parsing the semantics of user utterances, spoken language understanding (SLU) is a critical component of goal-oriented dialogue systems, which consists of two sub-tasks: intent detection and slot filling [Wang et al., 2005]. Recently, there have been massive efforts based on joint deep neural models [Zhang and Wang, 2016; Qin et al., 2020] to address the challenge of SLU in low-resource languages, where zero or minimal training data is available. To tackle the challenge of SLU in low-resource languages, we notice a fundamental opportunity – exploring structures of utterances. In general, given a user utterance, SLU detects the intent of the utterance and labels each word in the utterance if it belongs to a certain slot. Therefore, there is a natural hierarchical structure, utterance-slot-word, in the SLU task. This structure describes the complex relations between the intents and the slots. To improve the accuracy of a cross-lingual SLU system, it is crucial to capture multiple relations at different levels of granularity, which are ignored by all the existing multilingual code-switching methods [Liu et al., 2020a; Qin et al., 2020] that have shown superior performance in English, arguably the most resource-rich language. However, most of these methods demand large amounts of high-quality training data, which limits their applicability and effectiveness in low-resource languages where zero or minimal training data is available.

To address the lack of training data in low-resource languages, machine translation is applied to translate utterances in training sets in high-resource languages into low-resource languages [Chen et al., 2018; Upadhyay et al., 2018; Xu et al., 2020]. However, machine translators may not be available for some extremely low-resource languages [Upadhyay et al., 2018]. Therefore, code-switching approaches [Liu et al., 2020a; Qin et al., 2020] are developed to relieve the dependency on machine translators. The basic idea is to randomly select a set of words in the utterance to be replaced by the corresponding words in different languages through bilingual dictionaries without any machine translators. In this way, words in different languages become the context to each other whose representations are aligned in the universal semantic space automatically and implicitly. Code-switching methods have achieved promising results. But as shown in Table 1, although CoSDA-ML [Qin et al., 2020] substantially improves the slot filling performance, there still exists a challenging gap between the performance in high-resource languages such as English and that in low-resource languages.

| Method       | en   | es   | zh   | tr   |
|--------------|------|------|------|------|
| mBERT zero-shot | 95.90 | 77.80 | 56.99 | 47.43 |
| CoSDA-ML     | 95.62 | 84.44 | 79.48 | 53.46 |

Table 1: mBERT zero-shot and CoSDA-ML results (F1 score) on four languages of MultiATIS++ slot filling task.

\textsuperscript{*}Work done during the first author’s internship at Microsoft STCA.
\textsuperscript{†}Daxin Jiang and Xianglin Zuo are the corresponding authors.
previous methods.

In this paper, we propose a novel multi-level contrastive learning framework to learn the relations. First, at the utterance level, we develop a contrastive learning scheme to enhance the intent consistency of code-switched utterances (Figure 1(a)). Let $x_i$ be an utterance in a batch of source language training data. For each $x_i$, we generate the corresponding code-switched utterance $\hat{x}_i$. Although $\hat{x}_i$ is represented in mixed languages, it still has a meaning similar to $x_i$. Therefore, $\hat{x}_i$ is a positive example of $x_i$, while other instances ($x_j$ and $\hat{x}_j$, $j \neq i$) in the batch serve as the negative examples of $x_i$. To enhance the relation between the intent and the utterance, we maximize the difference of the distance in the representation space between $x_i$ and $\hat{x}_i$ with that between $x_i$ and the other instances in the batch.

Second, at the slot level, we model the relation between the slot values and the slots aggregating global information from all utterances (Figure 1(b)). Given each slot value in $x_i$, we select the the corresponding code-switched value in $\hat{x}_i$ as the positive example. Furthermore, we design a novel algorithm to generate hard negative examples, and a slot-guided value similarity function to facilitate the learning of relations between slot values and slot set.

Last, at the word level, we emphasize the relation between the words and their slot labels using local context in the utterance (Figure 1(c)). Each word of the slot is leveraged as the positive example of its slot label. We sample the words locally within the utterance as negative examples, which can be either labeled as other slot labels (type 1 negative) or out of any slots (type 2 negative). Applying contrastive learning on such positive/negative examples can strengthen the correlation between words and the slot labels (through type 1 negatives) and help the model better learn the slot value boundary (through type 2 negatives).

To further enhance the language transfer effectiveness, we propose a label-aware joint model concatenating the slot set with the utterance as the input to our model. This is motivated by the observation that, although the languages for utterances in cross-lingual SLU are diverse, the slot set is language invariant given a specific SLU task. By explicitly listing the slot set as the context for the utterances in different languages, the words and the slot set attend to each others’ representations in the model. The slots serve as “anchors” to align the words with similar semantic meaning in different languages.

Figure 1: Multi-level contrastive learning examples.

2 Related Work

2.1 Cross-lingual SLU

Cross-lingual SLU is a core component in dialogue systems that supports diverse languages. In general, most cross-lingual SLU methods fall into two categories: model transfer approaches and data transfer approaches.

The model transfer approaches target at learning cross-lingual embeddings that map the representations of different languages to a universal semantic space. Some representative methods are MUSE [Lample et al., 2018], CoVE [McCann et al., 2017] and pre-trained cross-lingual language models, such as mBERT [Devlin et al., 2019] and XLM-R [Conneau et al., 2020]. Based on the cross-lingual embeddings, an SLU model is trained on the source language data and then directly applied to target languages, especially low-resource languages [Upadhyay et al., 2018; Schuster et al., 2019; Li et al., 2021]. In addition to the general cross-lingual embeddings, [Liu et al., 2020b] further perform regularization by label sequence and adversarial training on the latent variable model [Liu et al., 2019] for alignment at the word and utterance levels.

The data transfer approaches focus on generating training data in target languages. In this direction, machine translation is widely adopted to translate training data from source languages to target languages, and is shown to be effective for improving model performance on low-resource languages [Upadhyay et al., 2018]. To reduce errors in word alignment for label transfer, [Xu et al., 2020] propose an end-to-end model to facilitate the utilization of translated data without alignment tools or methods [Schuster et al., 2019].

Our approach conducts both model transfer and data transfer. We employ cross-lingual models as the encoder, which belongs to the model transfer approach. We also apply code-switching [Qin et al., 2020] to generate positive and negative examples for contrastive learning, which carries the essence of the data transfer approach. Note that our data transfer approach only requires bilingual dictionaries to perform the code-switching without any machine translators.

2.2 Contrastive Learning

Contrastive learning [Saunshi et al., 2019] targets at learning effective representations by maximizing the similarity between the positive pairs, i.e., minimizing the distance in the feature space and keeping negative examples far from the
anchor example. In NLP, contrastive learning is firstly employed for learning sentence representations [Wu et al., 2020; Giorgi et al., 2021; Gao et al., 2021]. Recent studies extend this method to pre-trained cross-lingual language models. For example, [Chi et al., 2021] unify the cross-lingual pre-training objectives by maximizing mutual information and propose a sequence-level contrastive pre-training task. [Wei et al., 2021] conduct hierarchical contrastive learning based on XLM-R by randomly sampling negative examples at different levels.

The key difference between our work and the existing contrastive learning methods is that those methods focus on pre-training language models, while we consider the semantic structure in the SLU task and propose novel schemes to systematically generate hard negative examples at the utterance, slot, and word levels.

3 Methodology
Figure 2 illustrates our proposed framework. Specifically, we first propose a Label-aware Joint Model (LAJ) to transfer the semantics in the language-invariable slot set across different languages. Then, we develop a systematic approach of Multi-Level Contrastive Learning (MCL) by novel code-switching schemes at the utterance, slot, and word levels. The full framework is called LAJ-MCL.

3.1 Label-aware Joint Model
Given an utterance \(x = \{x_t\}_{t=1}^T\) with \(T\) words, denote by \(y^f\) and \(y^s = \{y^s_t\}_{t=1}^T\), respectively, the corresponding intent label and the slot label sequence. The network architecture for our label-aware joint model is shown in Figure 2(a). We adopt a pre-trained cross-lingual language model (mBERT, XLM-R, etc.) as the encoder \(M\). The input sequence consists of three parts: (1) the three special symbols \((s_O, s_B, s_I)\), the abstract labels representing outside-of-slot, beginning-of-slot, and inside-of-slot, respectively; (2) the slot set \(S = \{s_k\}_{k=1}^K\) corresponding to the \(K\) slots in the SLU task; and (3) the utterance \(\{x_t\}_{t=1}^T\). We concatenate the above three parts and add the special tokens [CLS] and [SEP]. The whole sequence is \(X = \{[CLS] s_O, s_B, s_I, [SEP] \{x_t\}_{t=1}^T\}\).

![Figure 2: The architecture of LAJ-MCL for cross-lingual SLU. In (b), we take the input \(X_0\) for illustration. The \(x_{0,j}\) and \(x_{0,t}\) denotes the slot value and word in the utterance \(x_0\) respectively.](image)

Embeddings for Slot Labels
We find that the text of slot labels often conveys some meanings. For example, the slot from loc in Figure 1(c) indicates that it is related to location names. Therefore, to initialize the embeddings for abstract slots \((s_O, s_B, s_I)\) and the slot set \(S\), instead of random initialization, we encode the slot labels by leveraging the semantics of their text descriptions through a pre-trained cross-lingual language model \(M\). We first feed the tokens of each slot label into the model and take the mean-pooling over the hidden states of the bottom 3 layers to obtain the token embedding. The mean-pooling over each token within the slot label is then utilized as the initial embedding. 

\(X\) is encoded by \(M\) to obtain the contextual embeddings \(E\) for the input utterance and slot labels, i.e., \(E = M(X)\), where \(E \in \mathbb{R}^{(5+K+T) \times d}\) is the representation matrix and \(d\) is the dimensionality of the hidden states of the encoder.

Label Compressor and Projector
For each slot \(s_k \in S\), there are two corresponding labels. \(B\)-\(s_k\) marks the beginning of the slot and \(I\)-\(s_k\) indicates that a word is inside the slot. To learn the representation for the BIO format slot labels, we design a label compressor to combine the abstract labels with the slots. First, the encoder result \(E_b\) for \(s_k\) is concatenated with the result for abstract labels, i.e., \(E_B\) for \(s_B\) and \(E_I\) for \(s_I\). Then, we feed them to the label compressor as \(E_b = (E_b \| E_B)W_{cb} + b_{cb}\) for \(B\) labels and \(E_i = (E_i \| E_I)W_{ci} + b_{ci}\) for \(I\) labels, where \(W_{cb}\) and \(W_{ci}\) are the weight matrices, and \(b_{cb}\) and \(b_{ci}\) are the bias vectors.

Before calculating the association between the words in an utterance and the slot labels, we apply a simplified projector similar to that in [Hou et al., 2020]. The words and labels are projected as \(H_i = E_iW_p + b_p\) and \(H_b = E_bW_p + b_p\), respectively, where \(W_p \in \mathbb{R}^{d \times d}\) is the weight matrix and \(b_p\) is the bias vector. Here, \(E_i \in \{E_O, \{E_B\}, \{E_I\}\}\), where \(E_O\) is the encoder output for \(S_O\), and \(\{E_B\}\) and \(\{E_I\}\) are the label compressor outputs for \(B\) labels and \(I\) labels. \(E_i\) is the encoder output for word \(x_i\). We push the model to learn a better representation so that the semantically related words and labels can be mapped close to each other.
Decoder for Intent Detection and Slot Filling

After applying the label compressor and projector, we predict the intent and slots for the utterance. For intent detection, we leverage the hidden state of $E_{CLS}$ and take $L_I = -y_i^I \log p_I$ as the loss function. Here $p_I = \text{softmax}(E_{CLS}W_I + b_I)$ is the intent classifier output, where $W_I$ and $b_I$ are the weight matrix and the bias vector, respectively.

For slot filling, the similarity between words and slot labels, $p_i^S = \text{softmax}(H_k \cdot [H_k]^T)$, is utilized for prediction. The loss function is formulated as $L_S = \sum_{t=1}^{T} -y_i^S \log p_i^S$, where $y_i^S$ is the slot label for word $x_t$. Last, the intent detection and slot filling are jointly optimized as:

$$L_J = L_I + L_S$$  \hspace{1cm} (1)

3.2 Multi-Level Contrastive Learning

Here, we propose a novel framework that employs the structural alignment between the source language and multiple target languages. Specifically, we apply contrastive learning at three levels, namely, the utterance, slot, and word levels, to capture complex relations, including those of intent and slot filling are jointly optimized as:

$$L_J = L_I + L_S$$  \hspace{1cm} (1)

Utterance-Level Contrastive Learning

As shown in Figure 1(a), for each source utterance $x_i$ in a batch, the corresponding code-switched instance $\hat{x}_i$ forms a positive example. All the other utterances and their code-switched instances are considered as the in-batch negative examples. Denote by $D = \{x_i\}_{i=1}^N$ a batch of the source language training data and by $\hat{D} = \{\hat{x}_i\}_{i=1}^N$ the code-switched counterpart, where $N$ is the batch size.

Word-Level Contrastive Learning

The two lists $A_k$ and $B_k$ are concatenated and go through a sentence embedding model to get $e_k$ to represent $s_k$.

To answer the second question, a basic method is to calculate the cosine similarity between the representations. However, there exists hard $x_{i,j}$ that are close to $x_{i,j}$ and $\hat{x}_{i,j}$ in the representation space but belong to different slots. Therefore, we introduce a slot-guided value similarity to focus on the slot-level semantics.

First, we apply mean-pooling to the encoder outputs of the slot value to obtain the slot representations $e_{i,j}$, $\hat{e}_{i,j}$ and $\hat{e}_{i,j}$. Then we evaluate the affinity of each slot value with respect to each slot, and finally calculate the value similarity by the KL-divergence with their affinity distribution. This method guides the model to pay attention to all the slots as the context information for the slot value and transfer the knowledge from the slots in the source language to multiple target languages.

To be more specific, given the representations of all slots $E^K = \{E_k\}_{k=1}^K$ from $\mathcal{M}$, let $E_K, e_{i,j}, \hat{e}_{i,j}$ and $\hat{e}_{i,j}$ go through the slot-level projection head $g_s(\cdot)$.

$$Z_s = g_s(\cdot) = \sigma(\cdot)W^s_1W^s_2$$  \hspace{1cm} (4)

where $\sigma$ is a ReLU activation function. Denote $Z^K_s = g_s(E^K)$, affinity $\text{aff}(z_i^s) = z_i^s(Z^K_s)^T$. Similarly, we can obtain the affinity $\text{aff}(\hat{z}_i^s)$ and $\text{aff}(\hat{z}_i^s)$. The slot-guided slot value similarity is

$$f_s(z_i^s, \hat{z}_i^s) = \text{KL}(\text{aff}(z_i^s), \text{aff}(\hat{z}_i^s))$$  \hspace{1cm} (5)

Finally, the slot-level contrastive loss with the margin $r_s$ is

$$L_s(x_{i,j}) = \max(0, f_s(z_i^s, \hat{z}_i^s) - f_s(z_{i,j}^s, \hat{z}_{i,j}^s) + r_s)$$  \hspace{1cm} (6)

Word-Level Contrastive Learning

Here we target the relation between words and their slot labels. Different from the slot-level method, which aggregating global information from all utterances, the word-level method considers the local context within an utterance. Given an input $x_{i,t}$ denote by $x_{i,t}$ the $t$-th word with label $y_{i,t}^S$. We consider each slot word as a positive example of its slot label. The negative examples can be sampled from the neighborhood of $x_{i,t}$ in the utterance. If the negative word belongs to
another slot label (type 1 negative), contrastive learning encourages the model to differentiate these different slot labels based on slot type (different slot) or label transition (same slot). And if the negative word does not belong to any slot, i.e., marked as O (type 2 negative), contrastive learning improves the model sensitivity to the slot value boundary.

To derive the negative examples $\bar{x}_{i,t}$, the words with the same slot label as $x_{i,t}$ are masked. Then, for each remaining word $x_{i,t}$, the probability $p_r$ of negative sampling based on the relative distance to $x_{i,t}$ is calculated as

$$p_r = \frac{q_r}{\sum_{r'} q_{r'}}$$

with $q_r = \sin\left(\frac{1}{r - \bar{r}}\right)$ (7)

We reuse the encoding for the slot labels and words by the label-aware joint model discussed before, i.e., output from the projector. Suppose $y_{i,t}^S = k$. The representations for $y_{i,t}$, $x_{i,t}$ and $\bar{x}_{i,t}$, i.e., $H_k$, $H_{i,t}$ and $\bar{H}_{i,t}$ go through the word-level projection head $g_w$ similar to $g_u$ and $g_s$, then obtain $x_{i,t}^w$, $\bar{x}_{i,t}^w$ and $\bar{z}_{i,t}^w$. The word-level contrastive loss is

$$\mathcal{L}_w(x_{i,t}) = \max(0, f_w(z_{i,t}^w, x_{i,t}^w) - f_w(z_{i,t}^w, \bar{z}_{i,t}^w) + r_w)$$

(8)

In addition, our word-level method is carried out on both source and code-switched utterances.

Finally, we derive the overall training loss of LAJ-MCL as

$$\mathcal{L} = \mathcal{L}_j + \lambda_1 \mathcal{L}_u + \lambda_2 \mathcal{L}_s + \lambda_3 \mathcal{L}_w$$

(9)

where $\lambda$’s are the hyper-parameters.

4 Experiments

In this section, we conduct a comprehensive evaluation of our proposed approaches on two benchmark datasets.

4.1 Experiment Settings

Datasets and Metrics

We experiment on two cross-lingual SLU benchmark datasets: MultiATIS++ [Xu et al., 2020] and MTOP [Li et al., 2021]. MultiATIS++ has 18 intents and 84 slots for each language and MTOP has in total 117 intents and 78 slots. To follow the zero-shot setting, we leverage the English training set (en) only for model training and evaluate on test set in all languages. The details of datasets are provided in Appendix.

We adopt the evaluation methods in the previous studies [Xu et al., 2020; Li et al., 2021], which compute the accuracy score and F1 score for intent detection and slot filling, respectively.

Implementation

We implement LAJ-MCL based on the pre-trained mBERT and XLM-R from [Wolf et al., 2020] as the encoder. Following [Qin et al., 2020], we use the bilingual dictionaries of MUSE for code-switching. The sentence replacement ratio is set to 1.0 and the word replacement ratio is set to 0.9. We train LAJ-MCL on the en training set augmented by code-switching and select the best checkpoint on the en validation set. More implementation details including hyper-parameters are described in Appendix.

4.2 Baselines

As we investigate the cross-lingual setting when no translation data for training is available, the main baseline methods include a zero-shot transferring method and code-switching.

ZJM. We re-implement the zero-shot joint model [Chen et al., 2019] (denoted as ZJM), which is trained on the en training set and directly applied to the target languages.

CoSDA-ML. [Qin et al., 2020] propose a dynamic code-switching method that randomly performs multilingual token-level replacement. For a fair comparison, we use both the en training set and code-switching data for fine-tuning.

4.3 Major Results

Tables 2 and 3 report the results of our method and the baseline methods on the two datasets. The “Model” column indicates the pre-trained cross-lingual language models employed as the utterance encoder. The results of ZJM and CoSDA-ML are derived from our re-implementation. Note that the performance of ZJM is equal to or even better than what reported the original papers [Xu et al., 2020; Li et al., 2021]. The results clearly show that LAJ-MCL significantly outperforms the baseline methods, especially for slot filling.

Table 2 shows the results on MultiATIS++, where we adopt both mBERT and XLM-Rbase as the base encoders in our approach. First, compared with the baseline method ZJM, both CoSDA-ML and our approach LAJ-MCL show large improvements because the code-switching (CS) method helps to better align the different languages. In a further investigation, we remove the CS method (w/o MCL&CS) and still observe clear gains over the ZJM method in slot filling (improved by 1.70 points). This finding proves the effectiveness of our LAJ approach, which leverages the language invariant slot set to align different languages.

Second, we compare LAJ-MCL with CoSDA-ML. Although both approaches apply code-switching, our method considers the semantic structure of the SLU task and develops novel code-switching schemes at the utterance, slot, and word levels. It shows 1.10 and 1.11 point gains in intent detection and 4.11 and 5.68 point gains in slot filling when using mBERT and XLM-Rbase, respectively. The improvement over CoSDA-ML verifies the effectiveness of our novel ideas that generate hard negatives for contrastive learning.

Besides the dramatic gains by the code-switching method over ZJM, our MCL significantly improves the slot filling performance on zh, ja, hi, and tr from the model perspective by explicitly aligning utterances, slot values, and words. Moreover, the average gain from CS of LAJ is larger than ZJM, which is attributed to introducing contextual label semantics. The replaced target language words are not only aligned with the source language words, but also attend to the representations of slot labels, which is a language adaptation process.

Following [Li et al., 2021], we further investigate the effectiveness of LAJ-MCL based on XLM-Rlarge, as shown in Table 3. We discover that our methods can still improve the intent accuracy and slot F1 score by 0.59 and 0.99 points, respectively. For both western and non-western target languages, LAJ-MCL obtains gains consistently compared with
The effectiveness of code-switching is shown in the third block and the results for multi-level contrastive learning are shown in the fourth block. The results show that each single level contrastive learning (+ UCL for utterance level, + SCL for slot level, and + WCL for word level) achieves some improvement compared with CS. Different level methods reveal different sensitivity to the target languages. For example, WCL improves the F1 score by 1.65 on tr while it has the lowest F1 score on ja. The UCL performs well on de and fr, but drops obviously on ja and tr. When combined in pairs as shown in the next three rows, the coupled CL methods make up the gap by using individual contrastive learning methods. Specifically, the UCL encodes the global semantics in an unsupervised manner, which causes coarse-grained alignment by the English utterance and code-switched utterance. The SCL and WCL leverage slot labels in the training data and perform fine-grained alignment within and between the languages. Consequently, the MCL framework achieves consistent improvements in intent detection and slot filling on the target languages.

### Effectiveness of Label-Aware Joint model

The purpose of the proposed label-aware joint model is to leverage the slot set to facilitate the attention between words and slots. Therefore, we primarily compare LAJ with [Chen et al., 2019] on different training data sizes in slot filling. As shown in Table 5, we observe that the proposed LAJ method shows consistent improvements over ZJM with respect to different training data sizes. The gain is large, especially when the train set is small. After applying code-switching, our model outperforms the baseline and increases the slot F1 score by a large margin.

### 5 Conclusion

In this paper, we target at the cross-lingual SLU scenario where no machine translator is available for the target language. We propose a Label-Aware Joint model (LAJ) and a novel Multi-level Contrastive Learning framework (MCL). The former leverages a language-invariable slot set to improve the alignment of languages while the latter exploits the
semantic structure of the SLU task and develops novel code-switching schemes to generate hard negative examples at the utterance, slot, and word levels. The results of extensive experiments verify the effectiveness of our approaches.

References

[Chen et al., 2018] Wenhu Chen, Jianshu Chen, Yu Su, Xin Wang, Dong Yu, Xifeng Yan, and William Yang Wang. Xi-nbt: A cross-lingual neural belief tracking framework. In Proc. of EMNLP, pages 414–424, 2018.

[Chen et al., 2019] Qian Chen, Zhu Zhuo, and Wen Wang. Bert for joint intent classification and slot filling. arXiv preprint arXiv:1902.10909, 2019.

[Chi et al., 2021] Zewen Chi, Li Dong, Furu Wei, Nan Yang, SakshamSingh, Wenhui Wang, Xia Song, Xian-Ling Mao, He-Yan Huang, and Ming Zhou. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. In Proc. of NAACL: HLT, pages 3576–3588, 2021.

[Conneau et al., 2020] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Proc. of ACL, pages 8440–8451, 2020.

[Devlin et al., 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proc. of ACL: HLT, pages 4171–4186, 2019.

[Gao et al., 2021] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. arXiv preprint arXiv:2104.08821, 2021.

[Giorgi et al., 2021] John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In Proc. of ACL-IJCNLP, pages 879–895, August 2021.

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

[Lample et al., 2018] Guillaume Lample, Alexis Conneau, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In ICLR, 2018.

[Li et al., 2021] Haoran Li, Abhinav Arora, Shuhuii Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. Mtot: A comprehensive multilingual task-oriented semantic parsing benchmark. In Proc. of EACL, pages 2950–2962, 2021.

[Liu et al., 2019] Zihan Liu, Jamin Shin, Yan Xu, Genta Indra Winata, Peng Xu, Andrea Madotto, and Pascale Fung. Zero-shot cross-lingual dialogue systems with transferable latent variables. In Proc. of EMNLP-IJCNLP, pages 1297–1303, 2019.

[Liu et al., 2020a] Zihan Liu, Genta Indra Winata, Zhaojiang Lin, Peng Xu, and Pascale Fung. Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems. In Proc. of AAAI, pages 8433–8440, 2020.

[Liu et al., 2020b] Zihan Liu, Genta Indra Winata, Peng Xu, Zhaojiang Lin, and Pascale Fung. Cross-lingual spoken language understanding with regularized representation alignment. In Proc. of EMNLP, pages 7241–7251, 2020.

[McCann et al., 2017] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: contextualized word vectors. In Proc. of NIPS, pages 6297–6308, 2017.

[Qin et al., 2020] Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP. In Proc. of IJCAI, pages 3853–3860, 2020.

[Qin et al., 2021] Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu. A co-interactive transformer for joint slot filling and intent detection. In ICASSP, pages 8193–8197, 2021.

[Saunshi et al., 2019] Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A theoretical analysis of contrastive unsupervised representation learning. In ICML, pages 5628–5637, 2019.

[Schuster et al., 2019] Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. Cross-lingual transfer learning for multilingual task oriented dialog. In Proc. of NAACL: HLT, pages 3795–3805, 2019.

[Upadhyay et al., 2018] Shyam Upadhyay, Manaal Faruqui, Gokhan Tür, Hakkani-Tür Dilek, and Larry Heck. (almost) zero-shot cross-lingual spoken language understanding. In ICASSP, pages 6034–6038, 2018.

[Wang et al., 2005] Ye-Yi Wang, Li Deng, and Alex Acero. Spoken language understanding. IEEE Signal Processing Magazine, 22(5):16–31, 2005.

[Wang et al., 2014] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In Proc. of CVPR, pages 1386–1393, 2014.

[Wei et al., 2021] Xiangpeng Wei, Rongxiang Weng, Yue Hu, Luxi Xing, Heng Yu, and WeiHua Luo. On learning universal representations across languages. In ICLR, 2021.

[Wolf et al., 2020] Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. Transformers: State-of-the-art natural language processing. In Proc. of EMNLP, pages 38–45, 2020.

[Wu et al., 2020] Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. Clear: Contrastive learning for sentence representation. arXiv preprint arXiv:2012.15466, 2020.
Algorithm 1: Generating Slot-level Negative Examples.

Input: English utterance \( x_i \), Code-switched utterance \( \tilde{x}_i \), Slot set \( S = \{ s_k \}_{k=1}^K \).

Output: Negative utterances \( \{ \{ \tilde{x}_i \} \} = N_s \).

1: for \( k = 1 \) to \( K \) do
2: \hspace{1em} Tokenize \( s_k \) into words and symbols to obtain \( A_k \)
3: \hspace{1em} Find the top-\( p_w \) frequent slot values of \( s_k \) to obtain \( B_k \)
4: \hspace{1em} Concatenating \( A_k \) and \( B_k \) as the input of MPNet provided by SentenceTransformers\(^1\) to obtain the representation \( e_k \) for \( s_k \)
5: end for
6: for \( k = 1 \) to \( K \) do
7: \hspace{1em} Select top-\( p_s \) similar slots for \( s_k \) by calculating the cosine similarity between the representations
8: \hspace{1em} \( \mathcal{V}_k = \{ \text{slot values } B_{k'} \text{ of each negative slot } s_{k'} \} \)
9: end for
10: for each slot value \( \tilde{x}_{i,j} \) in \( \tilde{x}_i \) do
11: \hspace{1em} Suppose the slot of \( \tilde{x}_{i,j} \) is \( s_k \)
12: \hspace{1em} Randomly sample \( N_s \) instances from \( \mathcal{V}_k \) as negative slot values
13: \hspace{1em} Replace \( \tilde{x}_{i,j} \) with code-switched negative slot values iteratively to generate \( \tilde{x}_{i,j} \)
14: end for

\(^1\)SentenceTransformers is an open-source library for sentence embeddings and language modeling.

Implement Details

We set the batch size to 32 and train the model for 20 epochs. We apply the AdamW optimizer, where the learning rate ranges from 1e-5 to 5e-5 with the linear scheduler. We select the best hyper-parameters by grid search for the margin \( r \)'s in triplet loss and loss coefficient \( \lambda \): \( r \)'s \{0.1, 0.3, 0.5, 0.7\}; \( \lambda \)'s \{0.3, 0.5, 0.7, 1.0\}. All the experiments are conducted on a NVIDIA A100 GPU.

Case Study

Table 8 lists several examples to illustrate the rationale behind our MCL method. In the first case, mittag means “noon” in English, and depart\_time\_time is the most frequently misclassified slot of depart\_time\_period\_of\_day according to our empirical study on the results of the baseline methods. Such errors can be addressed by our slot-level contrastive learning (CL) method, which replaces the words in a slot span with the words frequently in similar slots.

In the second case, más temprano means “earlier” in English. By word-level CL, the model reduces the error in slot boundary detection and changes from beginning-of-slot (B) to inside-of-slot (I).

For the last case, tacoma havaalani means “tacoma airport” in English. Our method learns to extend the slot value (through word-level CL). Moreover, the slot type is further corrected from city\_name to airport\_name, which can be attributed to slot-level CL. This case demonstrates the effectiveness of applying multi-level contrastive learning jointly.

A.2 Experiments

Datasets

MultiATIS++ (Table 6) is an extension of Multilingual ATIS. Human-translated data for six languages including es, de, zh, ja, pt, fr are added to Multilingual ATIS, which initially has hi and tr. There are 4478 utterances in the train set, 500 in the valid set, and 893 in the test set, with 18 intents and 84 slots for each language. MTOP (Table 7) is collected from the interactions between human and assistant systems. MTOP contains totally 100k+ human-translated utterances in 6 languages (en, de, es, fr, th, hi) across 11 domains. We use the flat version divided into 70:10:20 percentage splits for train, valid and test.
Table 6: Statistics of MultiATIS++

| Language      | Utterances | Intent types | Slot types |
|---------------|------------|--------------|------------|
|               | train      | valid | test |               |             |             |
| English-en    | 4488       | 490   | 893  | 18            | 84          |
| Spanish-es    | 4488       | 490   | 893  | 18            | 84          |
| Portuguese-pt | 4488       | 490   | 893  | 18            | 84          |
| German-de     | 4488       | 490   | 893  | 18            | 84          |
| French-fr     | 4488       | 490   | 893  | 18            | 84          |
| Chinese-zh    | 4488       | 490   | 893  | 18            | 84          |
| Japanese-ja   | 4488       | 490   | 893  | 18            | 84          |
| Hindi-hi      | 1440       | 160   | 893  | 17            | 75          |
| Turkish-tr    | 578        | 60    | 715  | 17            | 71          |

Table 7: Statistics of MTOP

| Number of utterances (train/valid/test) | Intent types | Slot types |
|----------------------------------------|--------------|------------|
| English-en                             | 22288        | 117        |
| German-de                              | 18788        | 78         |
| French-fr                              | 16584        |            |
| Spanish-es                             | 15459        |            |
| Hindi-hi                               | 16131        |            |
| Thai-th                                | 15195        |            |

Table 8: Case study of our MCL on MultiATIS++. **Bold** span in the case is the target slot value. **Red** and **Blue** indicate the false and true parts in the results, respectively.

| Case                                                                 | w/o MCL Result | Method   | MCL Result                              |
|----------------------------------------------------------------------|----------------|----------|-----------------------------------------|
| 1) Ich brauche Fluginformationen für einen Flug von Indianapolis nach Cleveland, der am Dienstag abfliegt | B-depart .time .time | + SCL    | B-depart .time .period .of .day          |
| 2) cuál es el vuelo más temprano entre baltimore y oakland con desayuno | B-flight .mod B-flight .mod | + WCL    | B-flight .mod I-flight .mod            |
| 3) tacom a takaal an , havalanidan sehir merkezine ulasim sagliyor mu ? | B-city .name O | + WCL&SCL | B-airport .name I-airport .name        |