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

Shining Liang[1,2,3]*, **Linjun Shou**[3], **Jian Pei**[4], **Ming Gong**[3],
Wanli Zuo[1,2], **Xianglin Zuo**[1,2]†, **Daxin Jiang**[3]†
[1]College of Computer Science and Technology, Jilin University
[2]Key laboratory of Symbolic Computation and Knowledge Engineering, MOE
[3]STCA, Microsoft
[4]Department of Electrical & Computer Engineering, Duke University
{liangsn17,zuoxl17}@mails.jlu.edu.cn; {lisho,migon,djiang}@microsoft.com; j.pei@duke.edu; zuowl@jlu.edu.cn

Abstract

Despite the great success of spoken language understanding (SLU) in high-resource languages, it remains challenging in low-resource languages mainly due to the lack of labeled training data. The recent multilingual code-switching approach achieves better alignments of model representations across languages by constructing a mixed-language context in zero-shot cross-lingual SLU. However, current code-switching methods are limited to implicit alignment and disregard the inherent semantic structure in SLU, i.e., the hierarchical inclusion of utterances, slots, and words. In this paper, we propose to model the *utterance-slot-word* structure by a multi-level contrastive learning framework at the utterance, slot, and word levels to facilitate explicit alignment. Novel code-switching schemes are introduced to generate hard negative examples for our contrastive learning framework. Furthermore, we develop a label-aware joint model leveraging label semantics to enhance the implicit alignment and feed to contrastive learning. 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

Spoken language understanding (SLU) is a critical component of goal-oriented dialogue systems, which consists of two subtasks: intent detection and slot filling (Wang et al., 2005). Recently, massive efforts based on the joint training paradigm (Chen et al., 2019; Qin et al., 2021) have shown superior performance in English. However, the majority of them require large amounts of labeled training data, which limits the scalability to low-resource languages with little or no training data. Zero-shot cross-lingual approaches have arisen to tackle this problem that transfer the language-agnostic knowledge from high-resource (source) languages to low-resource (target) languages.

Method	en	es	zh	tr
zero-shot	88.24	52.18	30.01	3.08
code-switching	88.69	54.42	45.24	7.41

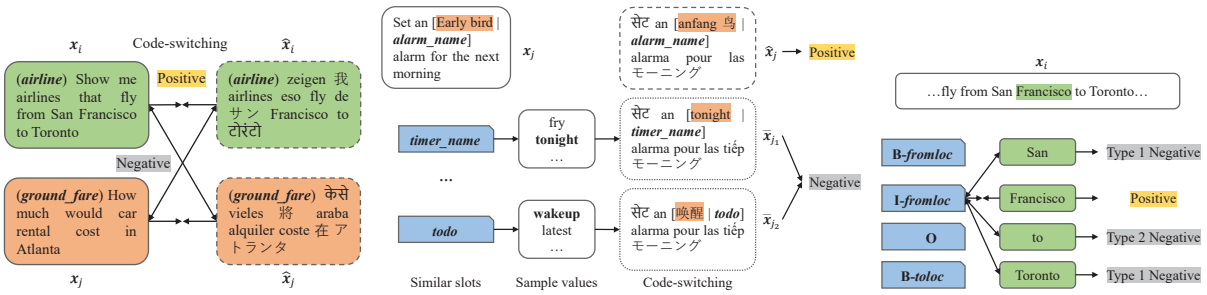
Table 1: mBERT based zero-shot and code-switching (CoSDA-ML) results on four languages of MultiATIS++ (semantic EM accuracy).

For the data-based transfer methods, machine translation is first applied to translate the source utterances into the targets (Upadhyay et al., 2018; Schuster et al., 2019; Xu et al., 2020). However, machine translation may be unreliable or unavailable for some extremely low-resource languages (Upadhyay et al., 2018). Therefore, multilingual code-switching (Liu et al., 2020a; Qin et al., 2020) is developed to reduce the dependency on machine translation, which simply uses bilingual dictionaries to randomly select some words in the utterance to be replaced by the translation words in other languages. Code-switching has achieved promising results as the word representations in the mixed-language context are aligned in a universal vector space, which is essential for cross-lingual transfer (Cao et al., 2020; Chi et al., 2021).

Despite the substantial improvements of CoSDA-ML (Qin et al., 2020) in Table 1, there still exists a challenging performance gap between English and the target languages. We believe only the implicit alignment of code-switching is insufficient for refining model representations. To address this issue, we advocate a fundamental methodology – exploiting structures of utterances. In general, given a user utterance, there is a natural hierarchical structure, *utterance-slot-word*, which describes the complex relations between the intents and the slots. To improve the transferability of a cross-lingual SLU system, it is crucial to employ multiple relations to

[1]Work is done during internship at Microsoft STCA.
[2]Corresponding authors.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 9903–9918
December 7-11, 2022



(a) Utterance Level (b) Slot Level (c) Word Level

Figure 1: Multi-level contrastive learning examples.

achieve explicit alignment at different levels, which is ignored by the previous methods.

In this paper, we propose a novel multi-level contrastive learning (CL) framework to perform explicit alignment. First, at the utterance level, we develop a CL scheme to enhance the intent consistency of code-switched utterances (Figure 1(a)). Let $\boldsymbol{x}_i$ be an utterance in a batch of source language training data. The corresponding code-switched utterance $\hat{\boldsymbol{x}}_i$ is its positive example as although $\hat{\boldsymbol{x}}_i$ is expressed in mixed languages, it has a similar meaning to $\boldsymbol{x}_i$. Other instances ($\boldsymbol{x}_j$ and $\hat{\boldsymbol{x}}_j$, $j \neq i$), meanwhile, serve as the in-batch negative examples of $\boldsymbol{x}_i$.

Second, at the slot level, we formulate the relation between the slot values and the slots by aggregating information from multiple utterances (Figure 1(b)). Given each slot value in $\boldsymbol{x}_i$, the corresponding code-switched value in $\hat{\boldsymbol{x}}_i$ is selected as the positive example. We design a slot-guided value similarity for CL, which leverages the probability distributions on the slot set of the slot values to achieve semantic alignment instead of computing similarity between them directly. Furthermore, we introduce a novel algorithm to generate hard negative examples from similar slot values.

Last, at the word level, we enhance the relation between the words and their slot labels using the context in an utterance (Figure 1(c)). Each word in the slot is a 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 CL on such positive/negative examples can strengthen the correlation between words and slot labels (through type 1 negatives) and help the model better learn the slot boundary (through type 2 negatives).

Moreover, we propose a label-aware joint model concatenating the slot set with the utterance as the input. This is motivated by the observation that, although the languages of utterances in cross-lingual SLU are diverse, the slot set is language-invariant. By listing the slot set as the context for the utterances in different languages, the words and the slot set can attend to each other's representations in the model. The slots are *implicit anchors* aligning the semantically similar words in different languages.

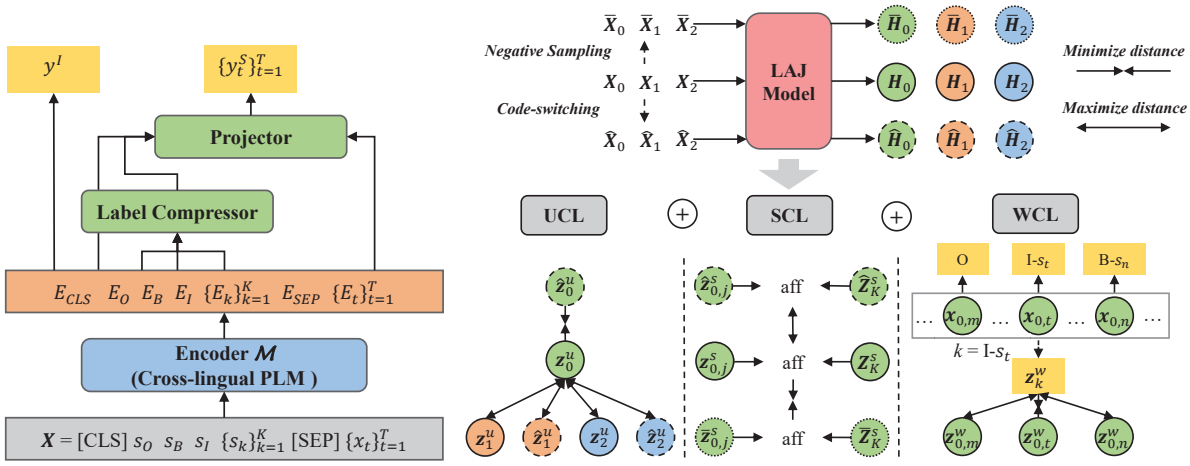
We conduct extensive experiments on two benchmark datasets. The experimental results show that the proposed label-aware joint model with a multi-level CL framework significantly outperforms strong baselines. Further analysis demonstrates the effectiveness of our method.

2 Related Work

2.1 Cross-lingual SLU

In general, most cross-lingual SLU methods fall into two categories: model-based transfer methods and data-based transfer methods.

The model-based transfer methods are based on cross-lingual word embeddings and contextual embeddings, such as are MUSE (Lample et al., 2018), CoVE (McCann et al., 2017) and cross-lingual pre-trained language models (PLMs) including mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020) and etc. A typical model-based transfer model is first fine-tuned on the source language data and then directly applied to target languages (Upadhyay et al., 2018; Schuster et al., 2019; Li et al., 2021a). More recently, additional components and training strategies have been developed. Liu et al. (2020b) perform regularization by label sequence and adversarial training on the latent variable model (Liu et al., 2019). van der Goot et al. (2021) propose three non-English auxiliary



(a) Label-aware Joint Model (b) Multi-level Contrastive Learning

Figure 2: The architecture of LAJ-MCL for cross-lingual SLU. In (b), we take the input $\boldsymbol{X}_0$ for illustration. The $\boldsymbol{x}_{0,j}$ and $\boldsymbol{x}_{0,t}$ denotes the slot value and word in the utterance $\boldsymbol{x}_0$ respectively.

tasks to improve cross-lingual transfer.

The data-based transfer methods focus on building training data in target languages. Machine translation is widely adopted to translate utterances from source languages to target languages and has been shown to effectively improve model performance (Schuster et al., 2019). Xu et al. (2020) propose a joint attention module for aligning the translated utterance to the slot labels to avoid label projection errors. As translated data has inherent errors and may be unavailable in low-resource languages, Liu et al. (2020a) and Qin et al. (2020) construct code-switching training data with bilingual dictionaries and fine-tune cross-lingual PLMs for implicit alignment.

2.2 Contrastive Learning

Contrastive learning (Saunshi et al., 2019) targets at learning example representations by minimizing the distance between the positive pairs in the vector space and maximizing the distance between the negative pairs. In NLP, CL is first applied to learn sentence embeddings (Giorgi et al., 2021; Gao et al., 2021). Recent studies have extended to cross-lingual PLMs. 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) and Li et al. (2021b) design a hierarchical CL by randomly sampling negative examples. Unlike the above methods using unlabeled parallel data for post-pretraining alignment, Gritta and Iacobacci (2021) and Gritta et al. (2022) utilize the `[CLS]` representation for utterance-level CL based on the translated data for task-specific alignment.

There is a contemporaneous work (Qin et al., 2022) that also proposes a multi-level CL method for explicit alignment. Similarly, we both take other utterances (source and code-switched) as negative examples at the utterance level. The differences are: (1) Qin et al. (2022) align the words while we focus on the slot label-word relation at the word level; (2) Qin et al. (2022) develop the alignment between the `[CLS]` and the words at the semantic level while we propose to generate negative slot values for the slot-level CL. Further comparison is in Appendix A.

3 Methodology

Figure 2 illustrates our method. On the left part, the proposed Label-aware Joint Model (LAJoint) transfers the semantics of the language-invariant slot set across languages. On the right part, we develop a systematic approach to Multi-level Contrastive Learning (MCL) by novel code-switching schemes. The full framework is called LAJ-MCL.

3.1 Label-aware Joint Model

Given an utterance $\boldsymbol{x} = \{x_t\}_{t=1}^{T}$ with T words, the corresponding intent label and the slot label sequence are y^I and $\boldsymbol{y}^S = \{y_t^S\}_{t=1}^{T}$, respectively. The architecture of our label-aware joint model is shown in Figure 2(a). We adopt a cross-lingual PLM as the encoder $\mathcal{M}$. The input sequence consists of three parts: (1) the three special symbols (s_O, s_B, s_I), i.e., the abstract labels representing

outside-of-slot, beginning-of-slot, and inside-of-slot, respectively; (2) the slot set $\mathcal{S} = \{s_k\}_{k=1}^{K}$ 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,

$$\boldsymbol{X} = \Big\{ \texttt{[CLS]}\, s_O, s_B, s_I, \{s_k\}_{k=1}^{K} \texttt{[SEP]} \{x_t\}_{t=1}^{T} \Big\} \quad (1)$$

Embeddings for Slot Labels We notice that the text descriptions of slot labels often convey specific meanings. For example, the slot `fromloc` in Figure 1(c) indicates that it's related to location names. Therefore, to initialize the embeddings of the abstract labels (s_O, s_B, s_I) and the slot set $\mathcal{S}$, the slot labels are encoded by leveraging the semantics of their text descriptions through the encoder. We first feed the tokens of each slot label to $\mathcal{M}$ and take the *mean-pooling* over the hidden states of the bottom 3 layers to obtain the token embedding. Then the normalized *mean-pooling* over each token within the slot label is utilized as the initial embedding.

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

Label Compressor and Projector There are two corresponding labels for each slot $s_k \in \mathcal{S}$. 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 of slot labels in the BIO format (Ramshaw and Marcus, 1999), we use a label compressor to combine the abstract labels with the slots. First, the encoder result $\boldsymbol{E}_k$ for s_k is concatenated with the result for abstract labels, i.e., $\boldsymbol{E}_B$ for s_B and $\boldsymbol{E}_I$ for s_I, then fed to the label compressor:

$$\boldsymbol{E}_k^B = (\boldsymbol{E}_B \| \boldsymbol{E}_k)\boldsymbol{W}_{cb} + \boldsymbol{b}_{cb} \quad (2)$$

$$\boldsymbol{E}_k^I = (\boldsymbol{E}_I \| \boldsymbol{E}_k)\boldsymbol{W}_{ci} + \boldsymbol{b}_{ci} \quad (3)$$

where $\boldsymbol{W}_{cb} \in \mathbb{R}^{2d\times d}$ and $\boldsymbol{W}_{ci} \in \mathbb{R}^{2d\times d}$ are weight matrices, and $\boldsymbol{b}_{cb}$ and $\boldsymbol{b}_{ci}$ are bias vectors.

Before calculating the association between the words and the slot labels, we further apply a projector similar to Hou et al. (2020) as below:

$$\boldsymbol{H}_t = \boldsymbol{E}_t\boldsymbol{W}_p + \boldsymbol{b}_p \quad (4)$$

$$\boldsymbol{H}_k = \boldsymbol{E}_k\boldsymbol{W}_p + \boldsymbol{b}_p \quad (5)$$

where $\boldsymbol{W}_p \in \mathbb{R}^{d\times d}$ is the weight matrix and $\boldsymbol{b}_p$ is the bias vector. Here, $\boldsymbol{E}_k \in \{\boldsymbol{E}_O, \{\boldsymbol{E}_k^B\}, \{\boldsymbol{E}_k^I\}\}$, where $\boldsymbol{E}_O$ is the encoder output for s_O, $\{\boldsymbol{E}_k^B\}$ and $\{\boldsymbol{E}_k^I\}$ are the label compressor outputs for B labels and I labels. $\boldsymbol{E}_t$ is the encoder output for word x_t. We hope the projector learns a better representation that the semantically related words and labels can be mapped close to each other.

Optimization For intent detection, we leverage the hidden state of $\boldsymbol{E}_{\texttt{CLS}}$ and take $\mathcal{L}_I = -y^I \log \boldsymbol{p}_I$ as the loss function. Here $\boldsymbol{p}_I = \mathrm{softmax}(\boldsymbol{E}_{\texttt{CLS}}\boldsymbol{W}_I + \boldsymbol{b}_I)$ is the intent classifier output, where $\boldsymbol{W}_I$ and $\boldsymbol{b}_I$ are weight matrix and bias vector, respectively. For slot filling, the similarity between words and slot labels, $\boldsymbol{p}_t^S = \mathrm{softmax}(\boldsymbol{H}_t\{\|\boldsymbol{H}_k\|_2\}^{\mathrm{T}})$, is utilized for prediction.The loss function is formulated as $\mathcal{L}_S = \sum_{t=1}^{T} -y_t^S \log \boldsymbol{p}_t^S$, where y_t^S is the slot label for word x_t. Last, the intent detection and slot filling are jointly optimized as:

$$\mathcal{L}_J = \mathcal{L}_I + \mathcal{L}_S \quad (6)$$

3.2 Multi-level Contrastive Learning

Here, we propose a novel framework that employs the semantic structure for explicit alignment between the source language and target languages. As shown in Figure 2(b), we apply CL at utterance, slot, and word levels to capture complex relations, including intent-utterance, slot-value, and slot label-word.

Denote by $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ a batch of the source language training data and by $\hat{\mathcal{D}} = \{\hat{\boldsymbol{x}}_i\}_{i=1}^{N}$ the code-switched data, where N is the batch size.

Utterance-level CL For each source utterance $\boldsymbol{x}_i$ in $\mathcal{D}$, the corresponding code-switched instance $\hat{\boldsymbol{x}}_i$ is its positive example. As shown in Figure 1(a), all the other source utterances and code-switched instances serve as the in-batch negative examples. We denote the negative example of $\boldsymbol{x}_i$ as $\bar{\boldsymbol{x}}_i$.

First, following prior studies (Wei et al., 2021; Qin et al., 2022), we take the encoder output of `[CLS]` as the utterance representation for $\boldsymbol{x}_i$, $\hat{\boldsymbol{x}}_i$ and $\bar{\boldsymbol{x}}_i$. Then, we map the representations to the contrastive space by the utterance-level projection head $g_u(\cdot)$:

$$\boldsymbol{z}^u = g_u(\cdot) = \sigma((\cdot)\boldsymbol{W}_1^u)\boldsymbol{W}_2^u \quad (7)$$

where $\cdot$ represents $\boldsymbol{e}_i$, $\hat{\boldsymbol{e}}_i$, and $\bar{\boldsymbol{e}}_i$, σ is the ReLU activation function. The purpose is to learn better representations for the following contrastive optimization and maintain more information in $\boldsymbol{e}$. Last,

the triplet loss (Wang et al., 2014) is adopted as the utterance-level contrastive loss:

$$\mathcal{L}_u(\boldsymbol{x}_i) = \max(0, f_u(\boldsymbol{z}_i^u, \hat{\boldsymbol{z}}_i^u) - f_u(\boldsymbol{z}_i^u, \bar{\boldsymbol{z}}_i^u) + r_u) \tag{8}$$

where the metric function f_u is L_2 distance and r_u is the loss margin.

Slot-level CL To conduct slot-level CL, an intuitive idea is to replace each slot value with the values that frequently appear in similar slots. In this way, the model learns to map the multilingual slot values to the corresponding slots in the vector space and differentiate values for different slots. We need to address the following questions. **Q1:** Given a slot s_k, how to define the *similar* slots and generate the negative examples? **Q2:** How to evaluate the *distance* between a slot value with both its code-switched positive and negative instance?

To answer **Q1**, we describe each slot s_k as its text description and the high-frequency slot values. For example, assuming that the slot s_k is `alarm_name`, it is tokenized into a list $\mathcal{A}_k$. Through the training data, we can identify some events frequently marked as `alarm_name`, among which the top-p_v frequent ones constitute a list $\mathcal{B}_k$. $\mathcal{A}_k$ and $\mathcal{B}_k$ are concatenated and fed to a sentence embedding model to get the embedding $\boldsymbol{e}_k$ for s_k. The similarity between the slots is then calculated by the cosine similarity between their representations. For each slot s_k, we define the set of hard negative values $\mathcal{V}_k$ as the union of $\mathcal{B}_{k'}$, where $\boldsymbol{s}_{k'}$ denotes the top-p_s similar slots with s_k.

Denote by $\boldsymbol{x}_{i,j}$ the j-th slot value in $\boldsymbol{x}_i$ and by $\hat{\boldsymbol{x}}_{i,j}$ the corresponding code-switched positive example in $\hat{\boldsymbol{x}}_i$. As shown in Figure 1(b), negative examples are derived by replacing each $\boldsymbol{x}_{i,j}$ by $\bar{\boldsymbol{x}}_{i,j}$ generated as follows. To maintain the consistency with the context of the positive example, the generation is conducted on $\hat{\boldsymbol{x}}_i$. Suppose the slot of $\hat{\boldsymbol{x}}_{i,j}$ is s_k, we sample a value from $\mathcal{V}_k$ and perform code-switching to get the hard negative example $\bar{\boldsymbol{x}}_{i,j}$. A negative utterance $\bar{\boldsymbol{x}}_i$ is then derived after replacing the values one by one. The generation algorithm is in Appendix B.1.

To answer **Q2**, a basic method is calculating the cosine similarity between the slot values. However, there exists hard $\bar{\boldsymbol{x}}_{i,j}$ that are close to $\boldsymbol{x}_{i,j}$ and $\hat{\boldsymbol{x}}_{i,j}$ in the vector 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 $\boldsymbol{e}_{i,j}$, $\hat{\boldsymbol{e}}_{i,j}$, and $\bar{\boldsymbol{e}}_{i,j}$. Second, we evaluate the affinity of each slot value with respect to each slot and calculate the value similarity by the KL-divergence of their affinity distributions. To be specific, given the representations of all slots $\boldsymbol{E}^K = \{\boldsymbol{E}_k\}_{k=1}^K$ from $\mathcal{M}$, let $\boldsymbol{E}^K$, $\boldsymbol{e}_{i,j}$, $\hat{\boldsymbol{e}}_{i,j}$ and $\bar{\boldsymbol{e}}_{i,j}$ go through the slot-level projection head $g_s(\cdot)$:

$$\boldsymbol{z}^s = g_s(\cdot) = \sigma((\cdot)\boldsymbol{W}_1^s)\boldsymbol{W}_2^s \tag{9}$$

where $\boldsymbol{Z}_K^s = g_s(\boldsymbol{E}^K)$. The affinity is defined as $\text{aff}(\boldsymbol{z}_{i,j}^s) = \boldsymbol{z}_{ij}^s(\boldsymbol{Z}_K^s)^{\mathrm{T}}$. Similarly, we can obtain $\text{aff}(\hat{\boldsymbol{z}}_{i,j}^s)$ and $\text{aff}(\bar{\boldsymbol{z}}_{i,j}^s)$. Then the slot-guided slot value similarity is formulated as:

$$f_s(\boldsymbol{z}_{i,j}^s, \hat{\boldsymbol{z}}_{i,j}^s) = \text{KL}(\text{aff}(\boldsymbol{z}_{i,j}^s), \text{aff}(\hat{\boldsymbol{z}}_{i,j}^s)) \tag{10}$$

This procedure guides the model to take the probability distribution on the slot set as the semantic information of the slot value and align this knowledge between the source language and the target languages. Finally, the slot-level contrastive loss with the margin r_s is:

$$\mathcal{L}_s(\boldsymbol{x}_{i,j}) = \max(0, f_s(\boldsymbol{z}_{i,j}^s, \hat{\boldsymbol{z}}_{i,j}^s) - f_s(\boldsymbol{z}_{i,j}^s, \bar{\boldsymbol{z}}_{i,j}^s) + r_s) \tag{11}$$

Word-level CL Unlike the slot-level method, which aggregates information from multiple utterances, the word-level method concentrates on the context within an utterance. Given an input $\boldsymbol{x}_i$, denote by $\boldsymbol{x}_{i,t}$ the t-th word with label $\boldsymbol{y}_{i,t}^S$. We consider each slot word as a positive example of its slot label. The negative examples can be sampled from the neighborhood of $\boldsymbol{x}_{i,t}$ in the utterance as shown in Figure 1(c). Suppose the negative word belongs to another slot label (type 1 negative). In this case, CL encourages the model to differentiate different slot labels based on slot type (different slot) or label transition (same slot). Furthermore, if the negative word does not belong to any slot, i.e., marked as O (type 2 negative), CL improves the model sensitivity to the slot value boundary.

To derive the negative examples $\bar{\boldsymbol{x}}_{i,t}$, the words with the same slot label as $\boldsymbol{x}_{i,t}$ are masked. For each remaining word $\boldsymbol{x}_{i,r}$, the negative sampling probability p_r is based on its relative distance to $\boldsymbol{x}_{i,t}$:

$$p_r = \frac{q_r}{\sum_{r'} q_{r'}} \text{ where } q_r = \sin(\frac{1}{|r-t|}) \tag{12}$$

We reuse the encoding of the slot labels and words from the projector in LAJoint model. Suppose $\boldsymbol{y}_{i,t}^S = k$. The representations for $\boldsymbol{y}_{i,t}^S$, $\boldsymbol{x}_{i,t}$

and $\bar{\boldsymbol{x}}_{i,t}$, i.e., $\boldsymbol{H}_k$, $\boldsymbol{H}_{i,t}$ and $\bar{\boldsymbol{H}}_{i,t}$, go through the word-level projection head g_w similar to g_u and g_s, then obtain $\boldsymbol{z}^w_k$, $\boldsymbol{z}^w_{i,t}$ and $\bar{\boldsymbol{z}}^w_{i,t}$. The word-level contrastive loss is:

$$\mathcal{L}_w(\boldsymbol{x}_{i,t}) = \max(0, f_w(\boldsymbol{z}^w_k, \boldsymbol{z}^w_{i,t}) - f_w(\boldsymbol{z}^w_k, \bar{\boldsymbol{z}}^w_{i,t}) + r_w) \tag{13}$$

where f_w is cosine similarity and r_w is the loss margin. 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 below:

$$\mathcal{L} = \mathcal{L}_J + \lambda_1 \mathcal{L}_u + \lambda_2 \mathcal{L}_s + \lambda_3 \mathcal{L}_w \tag{14}$$

where λ's are the hyper-parameters.

4 Experiments

4.1 Experiment Settings

Datasets and Metrics We conduct our experiments on two cross-lingual SLU benchmark datasets: MultiATIS++ (Xu et al., 2020) and MTOP (Li et al., 2021a). **MultiATIS++** has 18 intents and 84 slots for each language and **MTOP** has in total 117 intents and 78 slots. The details of the datasets are provided in Appendix C.1.

We adopt the evaluation metrics in the previous works (Xu et al., 2020; Li et al., 2021a) including intent detection accuracy, slot filling F1 score, and semantic exact match accuracy.

Implementation We build LAJ-MCL with the mBERT and XLM-R_{base} from Wolf et al. (2020) as the encoder. Bilingual dictionaries of MUSE are adopted for code-switching the same as Qin et al. (2020). Following the zero-shot setting, we use en training set and code-switching set for model training and en validation set for checkpoint saving. More details are described in Appendix C.2.

4.2 Baselines

We compare our model to the following baselines.

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

Ensemble-Net. Razumovskaia et al. (2021) propose an ensemble-style network whose predictions are the majority voting results of 8 trained single-source language models, which is a zero-shot model.

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 data and the code-switching data for fine-tuning.

GL-CLEF. Qin et al. (2022) propose a global-local contrastive learning framework for explicit alignment. It is a concurrent work of this paper.

4.3 Major Results

Table 2 shows the results on MultiATIS++. First, CoSDA-ML and our LAJ-MCL significantly outperform ZSJoint and Ensemble-Net as code-switching (CS) helps to align the representations across languages implicitly. Although both CoSDA-ML and LAJ-MCL apply code-switching, our framework considers the semantic structure of SLU and develops novel code-switching schemes in the multi-level CL. LAJ-MCL shows 21.7% and 8.8% improvements over CoSDA-ML on average semantic EM accuracy when using mBERT and XLM-R_{base} respectively, which verifies the effectiveness of leveraging multi-level CL for explicit representation alignment. Second, LAJoint performs better than ZSJoint and achieves greater gains with code-switching. Based on mBERT, LAJoint beats ZSJoint with 7.4% on EM accuracy, and it creases to 16.7% comparing *w/o MCL* with CoSDA-ML. It can be attributed that LAJoint introduces contextual label semantics. The replaced target language words are not only aligned with the source language words but also attend to the slot labels, which is a language adaptation process of the representations of slot labels.

In Table 3, we investigate the generalization of LAJ-MCL on MTOP with XLM-R_{base}. We find our methods can still improve the overall accuracy by 2.3%. The results demonstrate that our framework can scale out to multiple datasets and more languages.

For CL baselines, LAJ-MCL achieves similar results to GL-CLEF and achieve the SOTA performance based on code-switching on both datasets respectively. We leave extending our framework to translated data for future work.

4.4 Further Analysis

In this section, unless otherwise specified, all the methods use mBERT encoder on MutliATIS++.

Ablation Study To manifest the contribution of each component in LAJ-MCL, we conduct abla-

Data	Methods	mBERT			XLM-R_{base}		
		Intent Acc	Slot F1	Sem EM	Intent Acc	Slot F1	Sem EM
EN	ZSJoint*	87.00	68.08	38.02	90.94	66.79	38.85
	Ensemble-Net	87.20	55.78	-	-	-	-
	LAJoint (ours)	88.96	69.96	40.85	88.42	67.65	37.35
EN+CS	CoSDA-ML*	90.87	68.08	43.15	93.04	70.01	43.72
	GL-CLeF	91.95	**80.00**	**54.09**	**94.05**	74.81	46.35
	LAJ-MCL (ours)	**92.41**	78.23	52.50	93.49	**75.69**	**47.58**
	w/o MCL	92.01	76.11	50.37	91.86	75.33	46.41

Table 2: Average results of all the languages on MultiATIS++. Results with * are from our re-implementation. The full language breakdowns are shown in Appendix D.1.

Data	Methods	Intent Acc	Slot F1	Sem EM
EN	ZSJoint*	85.56	67.03	50.35
	LAJoint (ours)	82.61	64.22	46.55
EN+CS	CoSDA-ML*	90.72	73.34	58.77
	LAJ-MCL (ours)	**91.04**	**74.50**	**60.11**
	w/o MCL	90.67	73.61	58.92

Table 3: Average results of all the languages on MTOP. Results with * are from our re-implementation. The full language breakdowns are shown in Appendix D.1.

Methods	Intent Acc	Slot F1	Sem EM
LAJoint	88.96	69.96	40.85
- *Compressor*	89.21	69.64	40.01
- *Projector*	88.54	69.69	40.35
- *Comp&Proj*	87.95	68.74	39.90
LAJoint+CS	92.01	76.11	50.37
+ *UCL*	92.42	77.28	51.36
+ *SCL*	92.04	77.29	51.21
+ *WCL*	92.18	77.00	50.89
+ *UCL&SCL*	**92.57**	77.54	51.83
+ *UCL&WCL*	92.51	77.62	51.69
+ *SCL&WCL*	92.55	77.64	51.84
+ *MCL*	92.41	**78.23**	**52.50**

Table 4: Ablation study of difference components. *UCL*, *SCL*, and *WCL* denote utterance-level, slot-level, and word-level CL, respectively.

tion experiments, and the average results are in Table 15.

Both the label compressor and the projector play crucial roles in LAJoint. Intuitively, the label compressor learns the combination of abstract labels and slots, and then the projector learns to map the words closer to their corresponding slot labels. The average EM accuracy drops by 2.3% without additional layers (- *Comp&Proj*). When comparing LAJoint with ZSJoint, the clear improvements demonstrate the indispensability of our LAJoint model that leverages the language-invariant slot set to align representations implicitly.

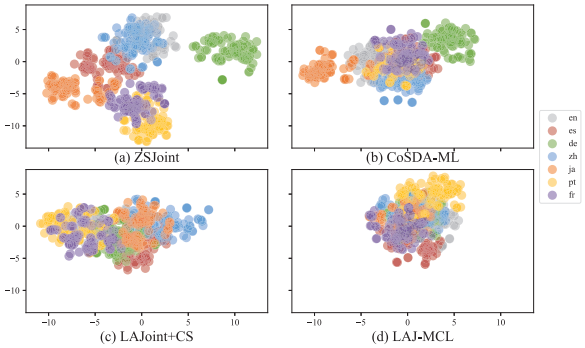


Figure 3: t-SNE visualization of utterance vector space. Dots in the same color denotes the utterance representations with the same intent.

Furthermore, we find that adding every single CL outperforms LAJoint+CS. Specifically, *UCL* encodes intent semantics and makes coarse-grained alignment between English and code-switched utterances by pulling the utterance representations close. Thus *UCL* achieves the highest intent accuracy among them. *SCL* and *WCL* leverage label semantics to perform fine-grained alignment across and within utterances by focusing on slot-value and slot label-word relations, respectively. When combined in pairs, the coupled CL outperforms the single CL, which demonstrates *UCL*, *SCL*, and *WCL* are complementary. Consequently, our MCL framework can achieve consistent improvements based on various cross-lingual PLMs.

The details about the full performance on all the languages are given in Appendix D.2.

Visualization To intuitively verify whether LAJ-MCL improves the alignment of model representations between languages, we select 100 parallel utterances with the same intent from all the test sets except hi and tr of MultiATIS++. It is hard to automatically extract parallel utterances with other languages since hi and tr utterances are fil-

Errors	es	de	zh	ja	pt	fr	hi	tr	Avg.
Method: CoSDA-ML									
#utterance	420	465	359	612	378	432	798	602	509.5
#slot_num	231	342	201	171	228	202	308	386	258.6
†*slot_type*	118	119	183	541	112	118	646	227	258.0
†*slot_bound*	47	15	6	12	27	85	48	44	35.5
†*slot_both*	106	55	47	102	97	131	70	86	86.8
Method: LAJoint+CS									
#utterance	337	289	346	538	357	369	720	581	442.1
#slot_num	159	179	186	147	215	208	296	331	215.1
†*slot_type*	106	108	176	472	87	66	512	243	221.3
†*slot_bound*	84	14	12	8	41	85	93	60	49.6
†*slot_both*	44	33	59	129	80	95	103	142	85.6
Method: LAJ-MCL									
#utterance	353	282	346	566	328	372	645	479	421.4
#slot_num	156	166	148	197	189	210	346	231	205.4
†*slot_type*	106	110	227	459	189	70	349	247	219.6
†*slot_bound*	86	15	13	4	36	75	52	70	43.9
†*slot_both*	77	45	34	85	101	110	78	59	73.6

Table 5: Error statistics of CoSDA-ML and our methods on the slot filling sub-task.

Hyper-params	p_v=0	p_v=10	p_v=20
p_s=5	49.78	50.91	**51.21**
p_s=10	49.75	51.05	51.15

Table 6: Slot-level CL hyper-parameters selection. We take LAJoint+SCL for example (Sem EM).

tered. Specifically, the encoder output of `[CLS]` is obtained as the utterance representation and visualized by t-SNE. The results are shown in Figure 3. It can be seen that there is only a small overlap between different languages in ZSJoint, i.e., the distance between different language representations is quite far. This problem is mitigated in CoSDA-ML where many dots overlaps, but there are some outliers. For our proposed methods, LAJoint+CS has better representation alignment than CoSDA-ML. In LAJ-MCL, the overlap region is further expanded and the internal distance of each language is reduced, which fully confirms the effectiveness of explicit alignment of our MCL framework.

Error Statistics We conduct error statistics on the prediction results of slot filling as shown in Table 5. In each block, *#utterance* is the number of utterances with errors, and *#slot_num* is the number of utterances in which the number of predicted slots are inconsistent with the ground truth. Furthermore, †*slot* errors are counted from the utterances without *#slot_num*. Specifically, †*slot_type*, †*slot_bound*, and †*slot_both* denote slot type error, slot value boundary error, and both errors, respectively. We have considered including the utterances with *#slot_num* errors in the statistics. However,

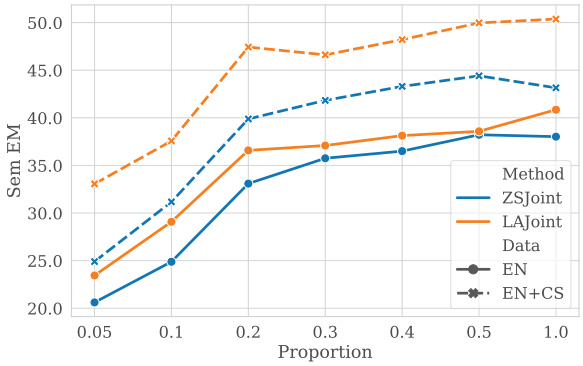


Figure 4: Comparison between ZSJoint and LAJoint by varying the proportion of training data (Sem EM).

when the model reduces *#slot_num*, the newly involved slot predictions affect the number of †*slot* errors, leading to confusing results.

The statistics show that: (1) Comparing CoSDA-ML and LAJoint+CS, the statistical results are almost consistent with the experimental results. By introducing label semantics, the average numbers of *#utterance* and *#slot_num* are significantly reduced by 13.2% and 16.8%, which proves that taking the slot set as the implicit anchor is effective for identifying slots in utterances on target languages. (2) LAJ-MCL continues to reduce the number of errors. Specifically, the average numbers of †*slot_bound* and †*slot_both* drop by 11.6% and 14.0%, which is exactly where LAJoint+CS did not improve compared with CoSDA-ML. Intuitively, †*slot_type* and †*slot_bound* errors benefit from slot-level and word-level CL, respectively. Our proposed SCL semantically enhances the relationship between slots and values, and WCL improves the the model's boundary detection and slots differentiation capability. In this way, both of them also contribute to reducing *#utterance*.

Hyper-parameters of SCL To evaluate our model's sensitivity to the negative examples generation related hyper-parameters of slot-level CL, i.e., top-p_s similar slots and top-p_v similar values, we conduct a grid-search experiment including the ranges: $p_s = \{5, 10\}$ and $p_v = \{0, 10, 20\}$ as shown in Table 6. We can observe that: (1) When not concatenating $\mathcal{B}_k$ for slot representation ($p_v = 0$), the average EM accuracy drops by 2.9% compared with $p_v = 20$ which confirms the effectiveness of high-frequency values; (2) Except for $p_v = 0$, the average EM accuracy is 51.08 ± 0.017 which indicates our slot-level CL is robust and not sensitive to the above hyper-parameters.

Methods	ATIS			SNIPS		
	Intent Acc	Slot F1	Sem EM	Intent Acc	Slot F1	Sem EM
Stack-Propagation (Qin et al., 2019)	97.50	96.10	88.60	99.00	97.00	92.90
Co-Interactive Trans (Qin et al., 2021)	98.00	96.10	88.80	97.10	**98.80**	**93.10**
SlotRefine (Wu et al., 2020)	97.74	96.16	88.64	99.04	97.05	92.96
SLG (Cheng et al., 2021)	98.30	**96.20**	88.70	**99.10**	97.10	**93.10**
LAJoint	**98.88**	96.08	**89.03**	99.00	97.02	93.00

Table 7: Results on ATIS and SNIPS. All the baseline results are from the original papers.

Effectiveness of LAJoint We conduct experiments to verify whether leveraging label semantics to facilitate the interaction between words and slots in LAJoint is effective. In Figure 4, LAJoint shows consistent improvements over ZSJoint with respect to different sizes and training data. The performance gap generally increases as the proportion decrease. Specifically, after applying code-switching (EN+CS), our model outperforms ZSJoint (i.e., CoSDA-ML) and increases EM accuracy by a large margin.

We further investigate whether LAJoint works for traditional SLU tasks, including AITS (Hemphill et al., 1990) and SNIPS (Coucke et al., 2018). Following the setting of current SOTA methods, we take $\text{BERT}_{\text{base}}$ (Devlin et al., 2019) as the encoder. The batch size is set to 32 and 64 for ATIS and SNIPS, respectively. The learning rate is selected from {5e-5, 6e-5, 7e-5, 8e-5, 9e-5} and the proportion of warm-up steps is 5%. Other details remain consistent with the multilingual experiment. As the results in Table 7, LAJoint shows competitive performance compared to both autoregressive (Qin et al., 2019, 2021) and non-autoregressive (Wu et al., 2020; Cheng et al., 2021) methods. As we don't incorporate the side information such as task-interaction (Qin et al., 2019, 2021) and sequential dependency (Cheng et al., 2021), it demonstrates that leveraging slot labels as the context of utterances is a simple and effective design.

5 Conclusion

In this paper, we propose a novel Label-aware Joint model (LAJoint) with a Multi-level Contrastive Learning framework (MCL) for zero-shot cross-lingual SLU. The former leverages the language-invariant slot set to transfer knowledge across languages and the latter exploits the semantic structure of SLU and develops contrastive learning based on novel code-switching schemes for explicit alignment. The results of extensive experiments demonstrate the effectiveness of our methods.

Limitations

The main contributions of this paper are towards aligning the representations of cross-lingual PLMs implicitly and explicitly by label semantics and multi-level contrastive learning. Our methods can be extend to other cross-lingual sequence labeling tasks. Nevertheless, we summarize two limitations for further discussion and investigation of the research community:

(1) *The improvement of LAJ-MCL on MTOP is not much significant as that on MultiATIS++.* MTOP has more intent labels than slot labels, and 6.5 times as many as MultiATIS++. We conjecture that it leads to a biased training process for LAJoint. In the future work, we plan to incorporate the intent labels to make full use of label semantics and achieve an unbiased training process.

(2) *The training and inference runtime of LAJ-MCL is larger than that of baselines.* The detailed results are in Table 10. We attribute the extra cost to the fact that LAJoint has longer input than ZSJoint, and LAJ-MCL dynamically generates negative examples in every batch. In the future work, we plan to design a new paradigm to replace the concatenation, thus reducing the requirement for GPU resources.

Acknowledgements

Shining Liang's research is supported by the National Natural Science Foundation of China (61976103, 61872161), the Scientific and Technological Development Program of Jilin Province (20190302029GX). Jian Pei's research is supported in part by the NSERC Discovery Grant program. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. In *International Conference on Learning Representations*.

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

Lizhi Cheng, Weijia Jia, and Wenmian Yang. 2021. An effective non-autoregressive model for spoken language understanding. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 241–250.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, He-Yan Huang, and Ming Zhou. 2021. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588.

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

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

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.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. Declutr: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895.

Milan Gritta, Ruoyu Hu, and Ignacio Iacobacci. 2022. Crossaligner & co: Zero-shot transfer methods for task-oriented cross-lingual natural language understanding. *arXiv preprint arXiv:2203.09982*.

Milan Gritta and Ignacio Iacobacci. 2021. Xeroalign: Zero-shot cross-lingual transformer alignment. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 371–381.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proc. of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1381–1393.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021a. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962.

Shicheng Li, Pengcheng Yang, Fuli Luo, and Jun Xie. 2021b. Multi-granularity contrasting for cross-lingual pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1708–1717.

Zihan Liu, Jamin Shin, Yan Xu, Genta Indra Winata, Peng Xu, Andrea Madotto, and Pascale Fung. 2019. Zero-shot cross-lingual dialogue systems with transferable latent variables. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1297–1303.

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

Zihan Liu, Genta Indra Winata, Peng Xu, Zhaojiang Lin, and Pascale Fung. 2020b. Cross-lingual spoken language understanding with regularized representation alignment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7241–7251.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, 30.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for 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 2078–2087.

Libo Qin, Qiguang Chen, Tianbao Xie, Qixin Li, Jian-Guang Lou, Wanxiang Che, and Min-Yen Kan. 2022. Gl-clef: A global–local contrastive learning framework for cross-lingual spoken language understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2677–2686.

Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu. 2021. A co-interactive transformer for joint slot filling and intent detection. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8193–8197. IEEE.

Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. 2020. Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3853–3860.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176.

Evgeniia Razumovskaia, Goran Glavaš, Olga Majewska, Anna Korhonen, and Ivan Vulic. 2021. Crossing the conversational chasm: A primer on multilingual task-oriented dialogue systems. *arXiv preprint arXiv:2104.08570*.

Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. 2019. A theoretical analysis of contrastive unsupervised representation learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5628–5637. PMLR.

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

Shyam Upadhyay, Manaal Faruqui, Gokhan Tür, Hakkani-Tür Dilek, and Larry Heck. 2018. (almost) zero-shot cross-lingual spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038. IEEE.

Rob van der Goot, Ibrahim Sharaf, Aizhan Imankulova, Ahmet Üstün, Marija Stepanović, Alan Ramponi, Siti Oryza Khairunnisa, Mamoru Komachi, and Barbara Plank. 2021. From masked language modeling to translation: Non-english auxiliary tasks improve zero-shot spoken language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2479–2497.

Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1386–1393.

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

Xiangpeng Wei, Rongxiang Weng, Yue Hu, Luxi Xing, Heng Yu, and Weihua Luo. 2021. On learning universal representations across languages. In *International Conference on Learning Representations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 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.

Di Wu, Liang Ding, Fan Lu, and Jian Xie. 2020. Slotrefine: A fast non-autoregressive model for joint intent detection and slot filling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1932–1937.

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.

A Related Work

Here we discuss in depth the differences from the contemporaneous work (Qin et al., 2022). To improve cross-lingual SLU task, although both (Qin et al., 2022) and we propose contrastive learning frameworks for explicit alignment, the details are the same only at the sentence (utterance) level. And the differences include: (1) For the token-level CL, GL-CLeF aligns each en token with its code-switched token across sentences and take all the other tokens as negative samples. Our MCL aligns slot labels with the corresponding tokens in the en and code-switched sentences respectively. We develop a negative sampling strategy to strengthen the correlation between slot labels and tokens and help the model better learn the slot boundary; (2) GL-CLeF introduces semantic-level CL aligns the `[CLS]` token with all the other tokens in the en and code-switched sentences respectively. While the slot-level CL of MCL focuses on the annotation targets: slot values. First, negative values pool is generated for each slot. Then, we propose the slot-guided value similarity based on label semantics and align slot values across the en and code-switched sentences.

B Method

B.1 Algorithm of Slot-level CL

The Algorithm 1 illustrates the process of generating negative examples in our slot-level CL. Here, p_v, p_s, and N_s are the hyper-parameters. For p_v and p_s, we empirically set them as 20 and 5, respectively. And for the negative examples, in order to balance the learning effectiveness and GPU memory usage, we set $N_s = 2$. In our word-level CL, for the negative examples $|\{\bar{\boldsymbol{x}}_{i,t}\}| = N_w$, we usually set $N_w = 2$.

C Experiment Settings

C.1 Datasets

MultiATIS++ is an extension of Multilingual ATIS (Table 8). Human-translated data for six languages including Spanish (es), German (de), Chinese (zh), Japanese (ja), Portuguese (pt), French (fr) are added to Multilingual ATIS which initially has Hindi (hi) and Turkish (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.

Algorithm 1 : Generating Slot-level Negative Examples.

Input: English utterance $\boldsymbol{x}_i$, Code-switched utterance $\hat{\boldsymbol{x}}_i$, Slot set $\mathcal{S} = \{s_k\}_{k=1}^{K}$.
Output: Negative utterances $|\{\bar{\boldsymbol{x}}_i\}| = N_s$.

for $k = 1$ to K **do**
  Tokenize s_k into words and symbols to obtain $\mathcal{A}_k$
  Find the top-p_v frequent slot values of s_k to obtain $\mathcal{B}_k$
  Concatenating $\mathcal{A}_k$ and $\mathcal{B}_k$ as the input of MPNet provided by SentenceTransformers [1] to obtain the representation $\boldsymbol{e}_k$ for s_k
end for
for $k = 1$ to K **do**
  Select top-p_s similar slots for s_k by calculating the cosine similarity between the representations
  $\mathcal{V}_k$ = [slot values $\mathcal{B}_{k'}$ of each negative slot $s_{k'}$]
end for
for each slot value $\hat{\boldsymbol{x}}_{i,j}$ in $\hat{\boldsymbol{x}}_i$ **do**
  Suppose the slot of $\hat{\boldsymbol{x}}_{i,j}$ is s_k
  Randomly sample N_s instances from $\mathcal{N}_k$ as negative slot values
  Replace $\hat{\boldsymbol{x}}_{i,j}$ with code-switched negative slot values iteratively to generate $\bar{\boldsymbol{x}}_{i,j}$
end for

MTOP is collected from the interactions between human and assistant systems (Table 9). MTOP contains totally 100k+ human-translated utterances in 6 languages (English (en), German (de), Spanish (es), French (fr), Thai (th), Hindi (hi)) across 11 domains. We use the flat version divided into 70:10:20 percentage splits for train, valid and test.

C.2 Implement Details

For code-switching, the sentence replacement ratio is set to 1.0 and the word replacement ratio is set to 0.9. We set the batch size N to 32 and train the model for 20 epochs. We apply the AdamW optimizer with the linear scheduler. We select the best hyper-parameters by grid search including the ranges: learning rate of the encoder {1e-5, 2e-5, 3e-5, 4e-5, 5e-5}; weight decay {0, 1e-3}; margin r's in triplet loss r's {0.1, 0.3, 0.5, 0.7}; loss coefficient λ's {0.3, 0.5, 0.7, 1.0}. The learning rate of the intent classifier and contrastive learning projection heads is 1e-3. The proportion of warm-up steps is 10%.

Language	Utterances			Intent types	Slot types
	train	valid	test		
en	4488	490	893	18	84
es	4488	490	893	18	84
pt	4488	490	893	18	84
de	4488	490	893	18	84
fr	4488	490	893	18	84
zh	4488	490	893	18	84
ja	4488	490	893	18	84
hi	1440	160	893	17	75
tr	578	60	715	17	71

Table 8: Statistics of MultiATIS++

Number of utterances (train/valid/test)						Intent types	Slot types
en	de	fr	es	hi	th		
22288	18788	16584	15459	16131	15195	117	78

Table 9: Statistics of MTOP

Following the zero-shot setting, we fine-tune the model on en training set and use en validation set for the hyper-parameters search. The best model checkpoint is decided by the semantic EM accuracy on en validation set. All the experiments are conducted on NVIDIA A100 and A6000 GPUs with NVIDIA's Automatic Mixed Precision. Our code is based on PyTorch and Transformers [2].

D Further Discussions

D.1 Full Major Results

The full comparison results on MultiATIS++ (Table 12, 13) and MTOP (Table 14).

D.2 Full Ablation Study

The full ablation results on MultiATIS++ are shown in Table 15. We observe that: (1) Different level CL methods show different sensitivity to the target languages. For example, *WCL* outperforms *SCL* on western languages, i.e., es, de, and fr, but significantly falls behind on ja, hi, and tr. In real scenarios, one can flexibly combine them according to the target languages. (2) The main performance improvement comes from the slot filling task. *MCL* shows 0.4% and 2.8% average improvements over LAJoint+CS on intent detection accuracy and slot filling F1 score, respectively. (3) Even though single CL, coupled CL, or MCL can not always perform better than LAJoint+CS on every target language, they achieve consistent improvement in average results of the three metrics, which indicates the generalization of our framework.

[2]https://github.com/huggingface/transformers

Methods	training	inference
ZSJoint	14	9
CoSDA-ML	15	9
LAJoint	18	16
LAJoint+CS	30	16
LAJ-MCL	65	16

Table 10: Comparison of training and inference runtime (second/epoch).

D.3 Case Study

Table 11 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 CL, 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 WCL). Moreover, the slot type is further corrected from `city_name` to `airport_name`, which can be attributed to SCL. This case demonstrates the effectiveness of applying multi-level contrastive learning jointly.

Case	w/o MCL Result	Method	MCL Result
(1) Ich brauche Fluginformationen für einen Flug von Indianapolis nach Cleveland , der am Dienstag **mittag** 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) **tacoma havaalani** , havalanindan sehir merkezine ulasim sagliyor mu ?	*B*-city_name *O*	+ *WCL&SCL*	*B*-airport_name *I*-airport_name

Table 11: Case study of our MCL on MutliATIS++. **Bold** span in the case is the target slot value. Red and Blue indicate the false and true parts in the results, respectively.

Intent Acc	en	es	de	zh	ja	pt	fr	hi	tr	**Avg.**
ZSJoint*	98.54	93.28	90.48	84.55	76.59	94.62	94.51	77.15	73.29	87.00
Ensemble-Net	90.26	96.64	92.50	84.99	77.04	95.30	95.18	77.88	75.04	87.20
LAJoint	98.54	96.30	93.17	89.25	83.31	95.41	95.97	82.53	66.15	88.96
CoSDA-ML*	97.98	95.07	95.07	91.04	85.67	95.18	95.97	84.88	76.92	90.87
GL-CLeF	98.77	97.05	97.53	87.68	82.84	96.08	97.72	86.00	83.92	91.95
LAJ-MCL	98.77	98.10	98.10	89.03	81.86	97.09	98.77	84.54	85.45	**92.41**
w/o MCL	98.32	97.87	97.31	91.60	86.67	96.86	97.76	88.58	73.15	92.01
Slot F1	en	es	de	zh	ja	pt	fr	hi	tr	**Avg.**
ZSJoint*	95.20	76.52	74.79	66.91	70.10	72.56	74.25	52.73	29.66	68.08
Ensemble-Net	85.05	77.56	82.75	37.29	9.44	74.00	76.19	14.14	45.63	55.78
LAJoint	95.80	80.69	76.63	67.24	74.47	72.20	76.23	54.22	32.12	69.96
CoSDA-ML*	95.32	80.82	79.63	80.40	65.69	79.30	79.21	49.29	50.53	73.36
GL-CLeF	95.39	85.22	86.30	77.61	73.12	81.83	84.31	70.34	65.85	**80.00**
LAJ-MCL	96.02	83.03	86.59	82.00	68.52	81.49	82.11	61.04	65.20	78.23
w/o MCL	95.39	85.85	86.13	81.35	69.45	81.17	82.42	54.01	49.24	76.11
Semantic EM	en	es	de	zh	ja	pt	fr	hi	tr	**Avg.**
ZSJoint	87.23	44.46	41.43	30.80	33.59	43.90	43.67	16.01	1.12	38.02
LAJoint	88.24	43.56	47.03	38.86	44.46	40.99	41.77	20.94	1.82	40.85
CoSDA-ML	87.23	50.28	45.35	55.10	26.32	55.21	49.38	8.29	11.61	43.15
GL-CLeF	88.02	59.53	66.03	50.62	41.42	60.43	57.02	34.83	28.95	**54.09**
LAJ-MCL	89.81	59.13	67.75	54.76	29.34	61.93	57.56	23.29	28.95	52.50
w/o MCL	87.46	61.03	66.97	56.44	34.83	58.68	57.56	16.35	13.99	50.37

Table 12: MultiATIS++ results as average Intent Detection Accuracy/Slot Filling F1 score/Semantic Exact Match Accuracy (mBERT encoder). Results with * are from our re-implementation.

Intent Acc	en	es	de	zh	ja	pt	fr	hi	tr	**Avg.**
ZSJoint*	98.99	98.10	92.83	87.91	84.43	93.95	97.09	87.12	78.04	90.94
LAJoint	98.77	92.72	89.81	86.45	81.97	93.73	92.61	84.66	75.10	88.42
CoSDA-ML*	98.99	98.99	98.32	89.70	83.76	97.98	98.66	89.70	81.26	93.04
GL-CLEF	98.66	98.04	98.43	91.38	88.83	97.76	97.85	93.84	81.68	**94.05**
LAJ-MCL	98.77	98.88	98.32	90.59	86.23	97.31	97.98	91.38	81.96	93.49
w/o MCL	98.88	98.88	97.98	84.99	85.11	93.62	98.54	89.14	79.58	91.86
Slot F1	en	es	de	zh	ja	pt	fr	hi	tr	**Avg.**
ZSJoint*	95.32	81.55	82.12	68.92	39.77	79.20	78.64	39.83	35.73	66.79
LAJoint	95.87	77.29	79.89	70.56	49.49	76.76	77.77	45.74	35.50	67.65
CoSDA-ML*	95.32	84.98	83.92	77.74	42.40	80.50	81.13	41.11	42.44	70.01
GL-CLEF	95.88	82.47	84.91	80.5	55.57	77.27	80.99	61.11	54.55	74.81
LAJ-MCL	95.87	83.10	83.88	79.55	64.30	79.31	81.43	54.96	58.80	**75.69**
w/o MCL	95.35	84.87	81.46	80.78	67.72	79.10	81.37	54.66	52.67	75.33
Semantic EM	en	es	de	zh	ja	pt	fr	hi	tr	**Avg.**
ZSJoint	88.24	52.18	57.89	30.01	4.59	54.20	52.41	7.05	3.08	38.85
LAJoint	88.91	43.23	49.94	31.13	12.43	49.27	47.93	11.31	1.96	37.35
CoSDA-ML	88.24	60.47	62.93	45.24	6.72	58.01	57.78	6.27	7.41	43.72
GL-CLEF	88.24	53.51	64.91	52.07	13.77	52.35	58.28	19.49	14.55	46.35
LAJ-MCL	88.58	57.22	55.99	53.75	27.88	55.10	55.66	12.09	21.96	**47.58**
w/o MCL	88.24	58.57	50.06	51.40	30.46	52.86	58.34	14.22	13.57	46.41

Table 13: MultiATIS++ results as average Intent Detection Accuracy/Slot Filling F1 score/Semantic Exact Match Accuracy (XLM-R_{base} encoder). Results with * are from our re-implementation.

Intent Acc	en	es	fr	de	hi	th	**Avg.**
ZSJoint	96.83	87.86	83.12	83.91	79.74	81.92	85.56
LAJoint	96.53	86.66	83.65	76.61	75.33	76.85	82.61
CoSDA-ML	96.92	94.16	91.23	92.76	80.75	88.50	90.72
LAJ-MCL	96.83	94.20	92.52	93.46	82.43	86.80	**91.04**
w/o MCL	96.85	93.86	91.79	92.76	81.93	86.84	90.67
Slot F1	en	es	fr	de	hi	th	**Avg.**
ZSJoint	91.88	70.93	72.94	67.16	49.88	49.37	67.03
LAJoint	91.58	68.60	68.46	61.14	49.72	45.85	64.22
CoSDA-ML	91.43	78.22	78.17	77.68	57.27	57.24	73.34
LAJ-MCL	91.78	78.11	78.16	78.35	61.79	58.82	**74.50**
w/o MCL	91.30	78.27	77.05	77.80	59.45	57.56	73.61
Semantic EM	en	es	fr	de	hi	th	**Avg.**
ZSJoint	84.65	54.70	52.02	46.18	32.20	32.33	50.35
LAJoint	84.34	49.90	46.45	39.19	31.52	27.92	46.55
CoSDA-ML	84.02	65.24	60.82	62.78	38.72	41.01	58.77
LAJ-MCL	84.59	65.68	63.98	63.57	40.59	42.28	**60.11**
w/o MCL	83.81	65.28	61.17	60.52	41.91	40.83	58.92

Table 14: MTOP results as average Intent Detection Accuracy/Slot Filling F1 score/Semantic Exact Match Accuracy (XLM-R_{base} encoder). Results with * are from our re-implementation.

Intent Acc	en	es	de	zh	ja	pt	fr	hi	tr	**Avg.**
LAJoint	98.54	96.30	93.17	89.25	83.31	95.41	95.97	82.53	66.15	88.96
-Compressor	98.43	96.98	89.59	88.13	83.87	94.40	94.96	81.86	74.69	89.21
-Projector	98.32	96.30	91.27	89.70	81.97	93.17	97.42	83.99	64.76	88.54
-Comp&Proj	98.10	96.19	93.17	88.24	84.77	96.08	95.86	84.43	54.69	87.95
LAJoint+CS	98.32	97.87	97.31	91.60	86.67	96.86	97.76	88.58	73.15	92.01
+UCL	98.43	97.65	96.98	90.93	88.13	96.98	98.54	87.12	77.06	92.42
+SCL	97.98	97.09	97.42	91.83	86.90	96.86	98.10	88.91	73.29	92.04
+WCL	98.32	97.65	97.54	90.03	88.58	96.64	97.98	87.35	75.52	92.18
+UCL&SCL	98.43	97.31	97.31	91.71	89.36	96.08	98.32	88.35	76.22	**92.57**
+UCL&WCL	98.43	97.98	97.54	90.15	85.67	96.75	98.10	86.45	81.54	92.51
+SCL&WCL	98.43	97.98	97.87	92.27	85.78	96.86	98.43	87.01	78.32	92.55
+MCL	98.77	98.10	98.10	89.03	81.86	97.09	98.77	84.54	85.45	92.41
Slot F1	en	es	de	zh	ja	pt	fr	hi	tr	**Avg.**
LAJoint	95.80	80.69	76.63	67.24	74.47	72.20	76.23	54.22	32.12	69.96
-Compressor	95.59	77.10	76.36	70.50	73.83	70.30	73.42	50.56	39.07	69.64
-Projector	95.75	76.18	77.56	69.21	71.82	74.90	76.64	49.98	35.18	69.69
-Comp&Proj	95.57	75.50	74.26	69.16	73.97	72.96	73.67	56.38	27.23	68.74
LAJoint+CS	95.39	85.85	86.13	81.35	69.45	81.17	82.42	54.01	49.24	76.11
+UCL	95.83	84.03	84.70	82.28	76.10	80.99	81.45	55.00	55.18	77.28
+SCL	95.61	84.22	85.11	81.26	76.20	81.25	81.79	56.58	53.63	77.29
+WCL	95.73	83.83	85.15	82.50	72.85	80.34	82.14	55.74	54.68	77.00
+UCL&SCL	95.81	85.71	84.72	82.23	73.82	81.70	81.41	57.10	55.35	77.54
+UCL&WCL	95.76	84.43	84.39	81.63	74.39	81.91	82.39	55.82	57.88	77.62
+SCL&WCL	95.85	83.15	83.69	81.45	73.48	80.60	78.35	62.07	60.12	77.64
+MCL	96.02	83.03	86.59	82.00	68.52	81.49	82.11	61.04	65.20	**78.23**
Sem EM	en	es	de	zh	ja	pt	fr	hi	tr	**Avg.**
LAJoint	88.24	52.74	50.84	34.60	43.23	42.89	39.31	18.48	2.10	40.85
-Compressor	88.24	44.34	46.25	36.95	44.68	40.31	37.96	15.45	5.87	40.01
-Projector	88.24	43.78	46.36	36.73	38.41	42.55	48.04	13.55	5.45	40.35
-Comp&Proj	88.58	40.87	44.34	37.63	44.23	42.67	37.07	21.50	2.24	39.90
LAJoint+CS	87.46	61.03	66.97	56.44	34.83	58.68	57.56	16.35	13.99	50.37
+UCL	88.58	59.91	65.96	55.43	40.20	58.23	52.97	21.39	19.58	51.36
+SCL	89.03	55.43	61.59	58.79	41.88	57.89	45.69	27.10	23.50	51.21
+WCL	88.80	58.23	65.29	58.45	39.53	56.33	57.78	15.45	18.18	50.89
+UCL&SCL	87.57	53.53	63.83	55.99	50.06	56.66	52.41	28.11	18.32	51.83
+UCL&WCL	88.69	55.77	66.29	55.54	44.01	58.68	55.88	21.84	18.18	51.69
+SCL&WCL	89.03	58.45	63.61	57.11	43.45	58.68	54.98	21.28	20.00	51.84
+MCL	89.81	59.13	67.75	54.76	29.34	61.93	57.56	23.29	28.95	**52.50**

Table 15: Ablation study of difference components on MutliATIS++ (mBERT encoder).