PRO-CS : An Instance-Based Prompt Composition Technique for Code-Switched Tasks

Srijan Bansal* **Suraj Tripathi*** **Sumit Agarwal***
Teruko Mitamura **Eric Nyberg**
{srijanb, surajt, sumita, teruko, en09} @andrew.cmu.edu
Language Technologies Institute, Carnegie Mellon University

Abstract

Code-switched (CS) data is ubiquitous in today's globalized world, but the dearth of annotated datasets in code-switching poses a significant challenge for learning diverse tasks across different language pairs. Parameter-efficient prompt-tuning approaches conditioned on frozen language models have shown promise for transfer learning in limited-resource setups. In this paper, we propose a novel instance-based prompt composition technique, PRO-CS, for CS tasks that combine language and task knowledge. We compare our approach with prompt-tuning and fine-tuning for code-switched tasks on 10 datasets across 4 language pairs. Our model outperforms the prompt-tuning approach by significant margins across all datasets and outperforms or remains at par with fine-tuning by using just 0.18% of total parameters. We also achieve competitive results when compared with the fine-tuned model in the low-resource cross-lingual and cross-task setting, indicating the effectiveness of our approach to incorporate new code-switched tasks. Our code and models will be available at https://github.com/srijan-bansal/PRO-CS

1 Introduction

Code-Switching (CS) is the phenomenon of shifting from one language to another in the same context, usually used in an informal way of speaking or writing (Sitaram et al., 2019; Jose et al., 2020). Within the scope of this paper, we have focused on the most commonly used form of code-switching which happens at the intra-sentential level. With the advent of social media platforms and the global rise in multilingual speakers, people generally converse in more than one language and often switch from one language to another (Parshad et al., 2016). Code-switching can go beyond the mere insertion of borrowed words, fillers, and phrases, and include morphological and grammatical mixing. It is frequently used to convey stronger emotions or to express one's ideas precisely (Sitaram et al., 2019).

Code-switching has gained attention in the NLP community due to a plethora of unstructured text data available that require an understanding of intermixing of two languages in the same sentence. Understanding the meaning of such text relies on

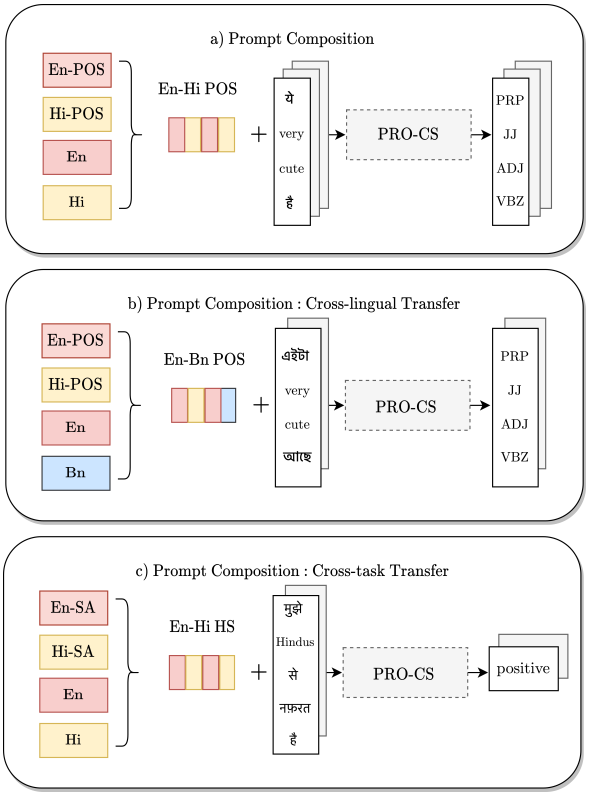


Figure 1: Different settings of PRO-CS [1], (a) prompt-composition for En-Hi POS code-switched task with source languages (En, Hi) and task (POS) prompts (b) prompt-composition for cross-lingual transfer to En-Bn POS from related source language Hi-POS prompt, (c) prompt-composition for cross-task transfer from En and Hi SA to En-Hi HS.

*Equal contribution

[1]Text in (a, b) He/She is very cute, (c) I hate Hindus. The languages shown are English (En), Hindi (Hi) and Bengali (Bn). The tasks used are Part of Speech (POS), Sentiment Analysis (SA), and Hate Speech (HS).

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 10243–10255
December 7-11, 2022

subtasks such as part-of-speech (POS) tagging, sentiment analysis (SA), and hate speech (HS), and the characteristic of CS languages. Recent work on CS tasks focuses on the identification of sentiments in online texts (Ravi and Ravi, 2016; Singh and Lefever, 2020), filtering of hate posts written in CS language (Ombui et al., 2019), part-of-speech tagging in CS text (Chandu et al., 2018; Chopra et al., 2021). Aguilar et al. (2020); Khanuja et al. (2020) have created benchmarks in a variety of code-switched language pairs and tasks.

One of the main challenges in the code-switched setting is the lack of high-quality labeled datasets across different language pairs and tasks. Recent works (Aguilar and Solorio, 2020; Sengupta et al., 2022; Khanuja et al., 2020) do not discuss much about the transferability of models to unseen combinations, which is essential for this low-resource domain. Parameter efficient approaches such as adapter-based models (Houlsby et al., 2019), prompt-tuning (Lester et al., 2021; Li and Liang, 2021) show promise for cross-lingual and cross-task transfer in multilingual few-shot and zero-shot settings (Pfeiffer et al., 2020a; Ansell et al., 2021a).

We take a step forward towards solving code-switching tasks through parameter efficient prompt-tuning with multilingual pre-trained language models (Devlin et al., 2019). Specifically, we wish to explore the effectiveness of the prompt-tuning approach for code switching by training prompts with multilingual LMs. Since code-switching comprises of information from two different languages along with the task information, we also investigate whether prompt-composition of task and language specific prompts can be applied for downstream CS tasks and how efficiently this method transfers to other code-switched language pairs or tasks. Essentially in this paper, we want to address three research questions.

- Are prompt-tuning methods that have been effective in monolingual and multilingual settings also effective in code-switched settings?
- Can we employ prompt-composition using task and language prompts for downstream code-switching task?
- How effective is prompt-composition in low-resource cross-lingual and cross-task settings?

Although current parameter-efficient techniques show promise for NLP, we find a big gap between the prompt-tuning and fine-tuning paradigms for code-switched NLP tasks. We feel this disparity is due to language and task complexities, as it involves switching between two languages. In this paper, we propose a novel PRO-CS prompt-composition technique that leverages language and task specific knowledge captured through language and task-based prompts. Task knowledge in prompts is learned through monolingual task corpora and language prompts are learned through masked token prediction via discriminative training on unlabeled language data.

All these prompts (referred as source prompts) serve as a pool of stored knowledge that can be reused for different language pairs and tasks. Unlike prompt-tuning, which learns prompts from scratch, we use an attention module to learn instance-based prompts by weighing the contributions of frozen source prompts and trainable prompts (referred as target prompts). We also use prompts trained on language identification, a characteristic task for code-switched language modeling to capture mixing between the two languages to initialize target language prompt. The frozen multilingual LM is then conditioned on the resulting prompt prepended with input for downstream tasks (see Section 3 for more details).

We hypothesize that both monolingual tasks and language understanding expose different facets of knowledge required for the code-switch setting. Our results support this claim, showing that our proposed approach outperforms prompt-tuning across all datasets. We test the generalizability of our approach by transferring across tasks and languages in low-resource setting (see Figure 1). For cross-lingual transfer, we substitute task prompt of a language (Bn-POS) with related language (Hi-POS). Similarly, for cross-task transfer, we replace task prompts (X-HS) with related tasks (X-SA).

In summary, the main contribution of our work are as follows (i) To the best of our knowledge, this is the first work investigating prompt-composition for code switching tasks. (ii) We leverage language and task-related prompts for PRO-CS and achieve consistent improvement over prompt-tuning. Further, we outperform or remain at par with the fine-tuning performance in most datasets with just 0.18% of the total parameters. (iii) We also show the effectiveness of PRO-CS for cross-lingual/cross-task transfer settings.

2 Background & Related Work

Fine-tuning pre-trained language models (Devlin et al., 2019; Yang et al., 2019) is arguably the most prevalent paradigm in NLP research for transfer learning (Ruder et al., 2019). In multilingual setting, fine-tuning language models trained on data from multiple languages has shown to be effective for downstream tasks and cross-lingual transfer (Devlin et al., 2019; Conneau et al., 2020).

Recent work in code-switched NLP (Khanuja et al., 2020; Winata et al., 2021) fine-tune large multilingual models for different downstream tasks. As the size of the pre-trained multilingual models (Devlin et al., 2019; Conneau et al., 2020) such as multilingual BERT (178M), XLM-R (470M), increases, it becomes computationally expensive to fine-tune them for ever evolving new tasks and new code-switching languages. In low-resource scenarios (such as code switching), fine-tuning these large pre-trained models (Chopra et al., 2021; Nayak and Joshi, 2022) is often susceptible to overfitting. Khanuja et al. (2020) attempts to mitigate this by generating synthetic CS data, which in most cases is not able to generate diverse examples to mimic real-world scenarios. Santy et al. (2021) shows that training multilingual LMs on real code-switched data is more helpful than training on synthetic data.

Code-switching datasets have very few annotated examples that present an ideal low-resource setting for parameter-efficient transfer learning. Recent study (Winata et al., 2021) shows promise in exploring parameter-efficient training (PET) for code switching using meta-embeddings that achieve similar performance to fine-tuned multilingual models with fewer parameters.

He et al. (2021) presents a unified view of different parameter-efficient training (PET) approaches like Adapter (Houlsby et al., 2019), Prompt-tuning (Lester et al., 2021), BitFit (Ben-Zaken et al., 2022), LoRA (Hu et al., 2021), and sparse fine-tuning (Guo et al., 2021) that have shown promise as an alternative to fine-tuning. These techniques are closely connected and share design elements that are essential for their effectiveness. We choose prompt-tuning in this study as it is being thoroughly explored in the NLP community and can be used as plug-ins for different tasks and language pairs with a frozen LM.

Recently, many studies have focused on PET approaches for multilingual settings. Hyper-X (Ustun et al., 2022) uses a hypernetwork to generate weights for adapter modules conditioned on both tasks and language embeddings. MAD-G (Ansell et al., 2021b) learns a single model to generate a language adapter for an arbitrary target language using the generation of contextual parameters. Zhao and Schütze (2021) show that discrete and soft prompting outperform fine-tuning in cross-lingual transfer. Polyglot prompt (Fu et al., 2022) proposes a two-tower encoder-based approach for language-independent prompt generation. Although PET techniques have been shown to be effective in monolingual and multilingual settings, this has not been explored in the code-switching domain.

Compositions of parameter-efficient methods for tasks and languages for better downstream transfer have been explored in the multilingual domain. MAD-X learns modular task and language-based adapters for cross-lingual transfer (Pfeiffer et al., 2020b). Ansell et al. (2021a) composes language and task information by selecting a subset of parameters that change the most during fine-tuning. It uses this composition to show zero-shot cross-lingual transfer. Asai et al. (2022) explores prompt-composition where the target prompt is learned by attending over multiple source task prompts. Motivated by this idea of prompt-composition and modular learning of task and language-based parameters for multilingual tasks, we explore the direction of prompt-composition for code-switching.

3 Method

Prompts can learn different facets of knowledge from both tasks and languages. In this paper, we hypothesize that monolingual task-based and language prompts can be used to compose instance-level code-switched task prompts. Such compositions are also compatible with heterogeneous supervision, that is, for different task and language combinations, and can be useful for initializing models in a low-resource setting.

3.1 Problem Setup and Motivation

We denote code-switched task as $\{t; cs(l_1, l_2)\}$ where t is the type of task, and cs is the code switched language comprising of two languages, l_1 and l_2. We use prompt-tuning (Liu et al., 2021) with frozen multilingual LM as a backbone model to first train source task prompts $\mathrm{P}^{t_1}_{l_i}, \mathrm{P}^{t_2}_{l_i}, ..., \mathrm{P}^{t_n}_{l_i}$ spanning source monolingual tasks $t_1, ..., t_n$ in language l_i (section 3.2) and source language prompts $\mathrm{P}_{l_1}, \mathrm{P}_{l_2}, ..., \mathrm{P}_{l_m}$ spanning across source

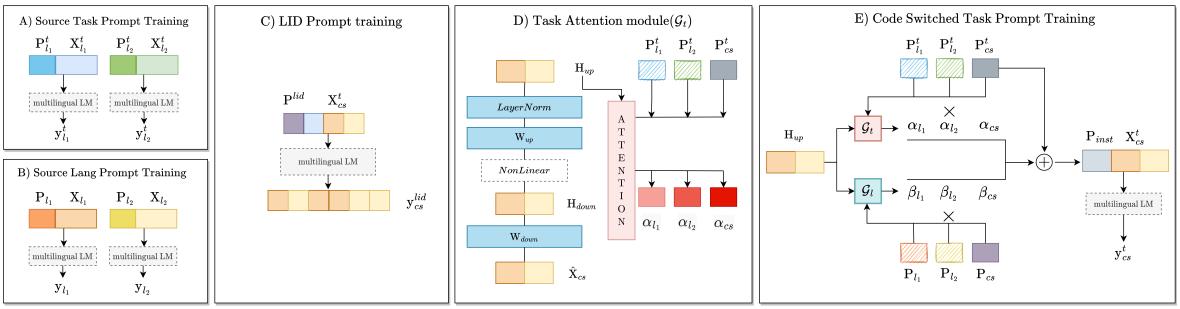


Figure 2: (A) Task prompts $\mathrm{P}^t_{l_1}, \mathrm{P}^t_{l_2}$ on source tasks in l_1, l_2 are learned through prompt-tuning. (B) Language prompts $\mathrm{P}_{l_1}, \mathrm{P}_{l_2}$ are learned via discriminative language modeling. (C) Language identification (lid) prompt training P^{lid} over code switched data which is used to initialise target language prompt P_{cs} (D) Task attention module $\mathcal{G}_t$ is learned to attend to the source task prompts $\mathrm{P}^t_{l_i}$ and target task prompt P^t_{cs}. Language module $\mathcal{G}_l$ (not shown) is trained in a similar way over language prompts. (E) P_{inst} is composed of source and target prompts (both language and task) which is trained on a frozen multilingual LM for downstream tasks. Dashed blocks and the multilingual LM are kept frozen during training of all of the above modules.

languages $l_1, ..., l_m$ (section 3.3). For the target code-switched task $\{t; cs(l_1, l_2)\}$, we learn both task and language-based target prompts (P^t_{cs}, P_{cs}) (section 3.4) along with separate attention module ($\mathcal{G}_t$, $\mathcal{G}_l$) (section 3.4.1) which attends over task prompts ($\mathrm{P}^t_{l_1}, \mathrm{P}^t_{l_2}, \mathrm{P}^t_{cs}$) and language prompts ($\mathrm{P}_{l_1}, \mathrm{P}_{l_2}, \mathrm{P}_{cs}$) respectively.

Given an input instance $(\mathrm{x}^t_{cs}, \mathrm{y}^t_{cs})$, we compose an instance-based prompt P_{inst} (section 3.4.2) by attending over the relevant source and target prompts. This P_{inst} is then prepended with the corresponding instance and passed to the frozen language model to learn all the trainable parameters. A detailed description of the architecture is shown in Figure 2. We now discuss each component of the model diagram in the following subsections.

3.2 Source Task Prompt Training

To learn source task prompts, we choose monolingual task corpora of tasks same as that of code-switching target tasks (like POS, SA). These prompts are trained using soft prompt-tuning (Liu et al., 2021) for each monolingual source task by keeping multilingual LM frozen.

Specifically, for source tasks t, a prompt $\mathrm{P}^t_l \in \mathbb{R}^{k \times d}$ is learned, where d is the hidden embedding size of the language model, and k is the number of prompt tokens. Prompts are initialized with random words from the vocabulary which has been shown to be more effective than initializing with random vectors (Li and Liang, 2021). Given an input $(\mathrm{x}^t_l, \mathrm{y}^t_l)$ for task t in language l, the input representation $\mathrm{X}^t_l \in \mathbb{R}^{n \times d}$ is generated by passing x^t_l (sequence of n tokens) through the embedding layer of the LM. The prompt is then prepended to this input $[\mathrm{P}^t_l; \mathrm{X}^t_l]$ and the model (parameterized by θ) is trained to predict the correct label y^t_l. Depending on the type of task, we apply a classification head on top of the [CLS] embedding (for classification), or use the last hidden state (for sequence tagging). During training, only prompt tokens are tuned to enable the prompt to capture the maximum amount of information related to the task at hand. The prompt parameters are trained exactly as fine-tuning, that is, by minimizing the classification loss $\mathcal{L}_C$ for the prediction of the output label (y^t_l):

$$\min_{\mathrm{P}^t_l} \mathcal{L}_C([\mathrm{P}^t_l; \mathrm{X}^t_l], \mathrm{y}^t_l, \theta)$$

3.3 Source Language Prompt Training

We train source language prompts $\mathrm{P}_l \in \mathbb{R}^{k \times d}$ for each source language l to capture language-specific knowledge. Inspired by discriminative pre-training of language models (Clark et al., 2020), we use a distilled mBERT as a generator to generate $\hat{\mathrm{X}}_l$ from masked input X^{MASK}_l using masked language modeling. X^{MASK}_l is obtained by randomly masking tokens from the input X_l. The mBERT discriminator learns to predict whether the tokens generated by the generator ($\hat{\mathrm{X}}_l$) match the input (X_l) or are fake, as shown in Figure 3. The generator is usually kept smaller than the discriminator to enable the discriminator to be better at capturing mistakes made by the generator. This discriminative pre-training has been shown to be better than masked-language modeling (Clark et al., 2020), and hence we apply the language prompt P_l in the discriminator concatenated with the input embedding of $\hat{\mathrm{X}}_l$. We train this model end-to-end with frozen discriminator model while keeping gener-

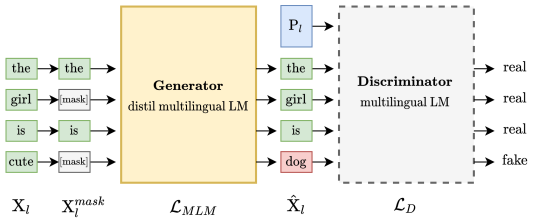


Figure 3: Discriminative source language prompt training with the generator being fine-tuned to generate $\hat{X}_l$ from masked input X_l^{mask} whereas the prompt in the frozen discriminator (dashed) learns to discriminate real tokens from fake ones.

ator and prompt tokens trainable. We minimize the weighted sum (denoted by λ) of MLM loss ($\mathcal{L}_{MLM}$) of the generator (θ_G) and the binary classification loss ($\mathcal{L}_D$) of the discriminator (θ_D). We set $\lambda = 50$ for our experiments.

$$\min_{P_l} \mathcal{L}_{MLM}(X_l, \theta_G) + \lambda \mathcal{L}_D([P_l; \hat{X}_l], \theta_D)$$

3.4 PRO-CS : Code-Switched Task Prompt Training

Given code-switched task $\{t; cs(l_1, l_2)\}$, we define two target prompts P^t_{cs} and P_{cs} to capture task and language specific information of code-switched task, respectively. Attention modules $\mathcal{G}_t$ and $\mathcal{G}_l$ are trained to learn the contribution of task and language source and target prompts. Learned attention weights are then used to compose an instance based $\{P_{inst}\}$ from target task prompt (P^t_{cs}), source task prompts $\{P^t_{l_1}, P^t_{l_2}\}$, target language prompt $\{P_{cs}\}$ and source language prompts $\{P_{l_1}, P_{l_2}\}$.

3.4.1 Training attention modules

Given input embedding $X^t_{cs} \in \mathbb{R}^{n \times d}$, and the pool of source and target prompts, PRO-CS's attention modules learn attention weights over these prompts per instance to compose instance-based prompt $P_{inst} \in \mathbb{R}^{k \times d}$ where k is the number of prompt tokens. We keep separate attention modules for task and language, $\mathcal{G}_t$ and $\mathcal{G}_l$ respectively, to enable the model to learn task-based information and language-based information separately.

Following (Asai et al., 2022), we model attention modules as bottleneck subnetworks (shown in Figure 2-D). Input X^t_{cs} is first max-pooled across sequence length such that $\hat{X}^t_{cs} \in \mathbb{R}^d$. This is then passed to sub-network that projects the pooled representation of the input to the prompt subspace.

We use $W_{down} \in \mathbb{R}^{r \times d}$ where ($r < d$) and $W_{up} \in \mathbb{R}^{d \times r}$ to project the input embedding $\hat{X}_{cs}$ to $H_{up} \in \mathbb{R}^d$. We use SILU as the nonlinear activation unit. This H_{up} is then used to calculate the attention scores on the source prompts $\{P^t_{l_1}, P^t_{l_2}, P_{l_1}, P_{l_2}\}$ along with the target prompts $\{P^t_{cs}, P_{cs}\}$.

$$\begin{aligned} H_{down} &= W_{down} \hat{X}^t_{cs} \\ H_{up} &= W_{up}(NonLinear(H_{down})) \\ H_{up} &= LayerNorm(H_{up}) \end{aligned}$$

Asai et al. (2022) computes logits by taking the dot product of H_{up} with the max-pooled representation of P_i. However, max-pooling leads to information loss, resulting in similar attention scores. We address this by summing k-dimensional vector obtained by product ($P_i H_{up}$) where $P_i \in \mathbb{R}^{k \times d}$ and $H_{up} \in \mathbb{R}^d$. We use similar architectures for the task and language attention modules $\mathcal{G}_t$ and $\mathcal{G}_l$ to compute H^t_{up} and H^l_{up} respectively. $P_i \in \{P^t_{l_1}, P^t_{l_2}, P^t_{cs}\}$ is used for $\mathcal{G}_t$ while $\mathcal{G}_l$ uses $P_j \in \{P_{l_1}, P_{l_2}, P_{cs}\}$.

$$a_i = \sum_k (P_i H^t_{up}) \qquad b_j = \sum_k (P_j H^l_{up})$$

We apply softmax over logits to compute attention weights, α_i for task prompts and β_j for language prompts. We make use of the softmax temperature introduced by Radford et al. (2021) and scale the logits by $1/d \times \exp(K)$ to prevent the attention modules from being overconfident, where K is a hyperparameter and is set to 1 for our experiments.

$$\alpha_i = \frac{e^{a_i}}{\sum_{i \in \{l_1, l_2, cs\}} e^{a_i}} \qquad \beta_j = \frac{e^{b_j}}{\sum_{i \in \{l_1, l_2, cs\}} e^{b_j}}$$

3.4.2 Instance-based prompt-composition

The learned attention weights over source task prompts, source language prompts, target task, and language prompt are used to compose the instance-based prompt (P_{inst}). We explicitly add the target prompts to the weighted sum to ensure the influence of target prompts in the composition. Thus target prompts will always be updated even if their corresponding attention weights are very small.

$$\begin{aligned} P_{inst} = (&P^t_{cs} + \alpha_{l_1} P^t_{l_1} + \alpha_{l_2} P^t_{l_2} + \alpha_{cs} P^t_{cs}) + \\ &(P_{cs} + \beta_{l_1} P_{l_1} + \beta_{l_2} P_{l_2} + \beta_{cs} P_{cs}) \end{aligned}$$

Similarly to the original task-based prompt training, this instance-based prompt P_{inst} is then used along with X^t_{cs} to learn the CS classification/sequence tagging task by optimizing the loss $\mathcal{L}_{CS}$ of predicting the label y^t_{cs}.

$$\min_{P_{inst}} \mathcal{L}_{CS}([P_{inst}; X^t_{cs}], y^t_{cs}, p_\theta)$$

3.4.3 Target prompt initialization

Language identification (LID) is the task of identifying words from languages l_1 and l_2 in code-switched data and capturing the notion of shift from one language to another. Various code-switching models have modeled downstream CS tasks with language identification as an auxiliary task (Chandu et al., 2018), which helps the model by learning code-switching points. Inspired by that, we initialize our target language prompt P_{cs} by a prompt trained on the LID task for the code-switched language pair to add inductive bias for the downstream task. The target task prompt P^t_{cs} is initialized with words from the vocabulary, similar to source task prompt training and is kept trainable. The remaining source prompts (both language and task) are kept frozen.

4 Datasets

We evaluate the performance of our model on a diverse set of code-switching datasets that span a variety of tasks and language pairs.

4.1 Code-switched datasets

We make use of datasets used in popular code-switching benchmarks, GLUECoS (Khanuja et al., 2020) and LinCE (Aguilar et al., 2020). In this study, we focus on classification and sequence tagging datasets due to their availability across different language pairs. For classification, we use sentiment analysis (SA), hate speech (HS), and intent classification (IN) datasets, and use part-of-speech tagging (POS), named entity recognition (NER) datasets for sequence tagging. These datasets span across multiple domains and the cardinality of their label set varies across language pairs.

We run experiments on 10 datasets across 4 language pairs. Details about the dataset and their sources are mentioned in Table 1. We evaluate our model in full data setting on tasks in En-Hi and En-Es as these two language pairs are most ubiquitous in the real world. We simulate a low resource setting for code-switched tasks which do not have relevant source monolingual task data. We evaluate cross-lingual and cross-task transfer in low-resource setting to test transferability of our proposed prompt-composition.

4.2 Monolingual datasets

Source task prompts : Monolingual datasets used for training source task prompts are discussed in appendix (see Table 4). Similar to our target task setting, source tasks consist of classification and sequence tagging corpus across different languages. For languages whose monolingual source task does not exist, we choose relevant task in another language to show cross-lingual transfer.

Corpus	Lang	#train	#dev	#labels
Sentiment Analysis (C)	En-Hi	14k	3k	3
	En-Es	12k	3k	3
	En-Ta	11k	1.2k	4
Hate Speech (C)	En-Hi	2.7k	0.9k	2
Intent (C)	En-Hi	27k	3k	22
Part of Speech (S)	En-Hi	2k	0.2k	44
	En-Es	2.1k	0.2k	36
	En-Bn	0.5k	0.06k	39
Named Entity Recog (S)	En-Hi	1.2k	0.3k	7
	En-Es	33k	10k	19

Table 1: Code-switched datasets across different target tasks and language pairs, where C denote classification and S denote sequence tagging tasks.

Source language prompts : We used raw Wikipedia data for source language prompt training using discriminative language modeling. We took the latest Wikipedia dump and used WikiExtractor[2] to extract 100k examples per language for training.

5 Experiment

We describe the baselines used in this paper and the experimental setup of our PRO-CS approach in the following subsections. We also discuss the transfer learning settings on which we evaluate our model.

5.1 Baselines

We evaluate our proposed technique, PRO-CS against fine-tuned multilingual BERT (178M trainable parameters), denoted as FT. Furthermore, we adopt the popular prompt-tuning technique (Lester et al., 2021) denoted as PT to compare with the proposed prompt-composition approach. Prompt-tuning has been shown to generate competitive results on the popular SuperGLUE (Wang et al., 2019) benchmark, but has not been explored in the code switch setting.

[2]https://github.com/attardi/wikiextractor

		full-data						*cross-lingual** (Hi → X)		*cross-task** (SA → X)	
		POS		NER		SA		POS	SA	HS	IN
Setting	Params	En-Hi	En-Es	En-Hi	En-Es	En-Hi	En-Es	En-Bn	En-Ta	En-Hi	En-Hi
FT	178M	**60.76**	81.14	**75.09**	**49.93**	66.06	42.15	**65.75**	28.38	**63.12**	43.74
PT	77K	54.43	77.99	70.94	45.98	63.34	43.00	62.33	25.38	61.33	39.95
PRO-CS	314K	60.33	**82.98**	71.98	46.02	**67.88**	**45.52**	64.30	**29.57**	62.82	**45.03**

Table 2: Macro F1 scores for different classification and sequence tagging tasks in En-Hi, En-Es for fine-tuning (FT), prompt-tuning (PT) and our PRO-CS in the full-data setting. It also compares the models in cross-lingual (Hi → X) and cross-task settings (SA → X) using 512 examples that simulate low-resource scenarios

5.2 Experimental setting

For code-switching tasks, the relevant source task and language prompts are chosen for our proposed prompt composition method (PRO-CS). We run experiments on `bert-base-multilingual-cased` [3] as our base model. Due to the unavailability of test set in most datasets, we use the development set of each corpus to report our model's performance. We create a small subset from the training set (known as validation set) for hyperparameter tuning and checkpoint selection. After extensive tuning, we keep learning rate of 1e-3, batch size of 32 and maximum number of tokens as 512 for all the experiments. Since, the label distribution in datasets is very unbalanced (specially for sequence tagging), we report the macro F1 score (mean of the F1 score for each label). All models in target code-switch tasks are trained for 20 epochs, and the best checkpoint based on validation metric is used for evaluation. Both the source task and language prompts are trained for 40 epochs on monolingual datasets. All prompts consist of $k = 100$ tokens with $d = 768$. We run our experiments on a single 3090 GPU.

Transfer Learning : We evaluate cross-lingual and cross-task transferability of different approaches on CS tasks which do not have relevant source monolingual task data available. In such cases, we transfer knowledge from high-resource code-switch tasks by using models trained on them to initialise models for low-resource task. This includes backbone model for finetuning, prompts for prompt-tuning and target prompts and attention modules for PRO-CS. We further replace the source task prompt with related prompt from our pool of monolingual source prompts for PRO-CS. For cross-lingual transfer, we substitute the non-English source task prompt with prompts trained on same task for a high-resource language. Similarly, for cross-task transfer, we use source task prompt of the most relevant task of the same language pair. (Refer Figure 1)

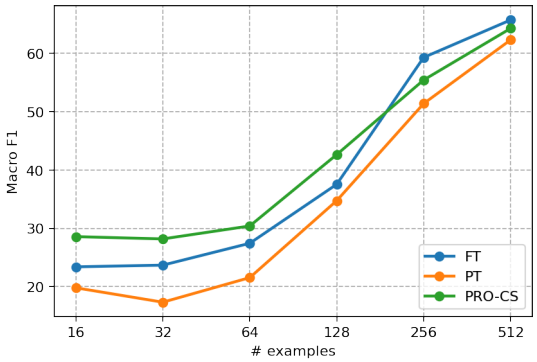
We evaluate both the cases of transfer learning in low resource setting. The low-resource setting aptly mimics the real-world scenario for CS domain, since downstream tasks often have much lower-quality data. We train models on small subsets of 512 training instances (low-resource) that are sampled such that the label distribution is balanced. We evaluated these models on the entire development set.

6 Result

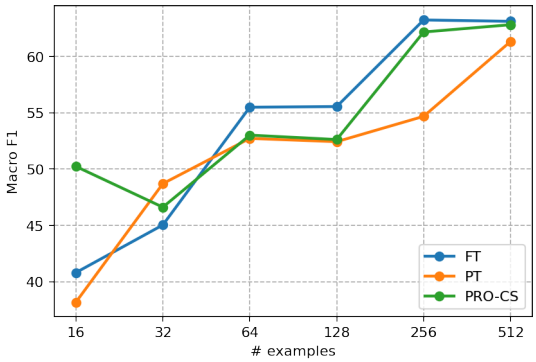
Table 2 compares the performance of our prompt-composition approach (PRO-CS) with fine-tuning (FT) and prompt-tuning (PT) on code-switching tasks. We see that our model outperforms prompt-tuning by a significant margin across all datasets, showing that language and task based composition is more effective in code switching than prompt-tuning. On average, we achieve an improvement of 3 and 3.5 points in Macro-F1 scores on sequence tagging and classification tasks compared to the prompt-tuning approach in the full data setting.

We achieve competitive results compared to fine-tuning setting and even outperform it for En-Es SA, En-Es POS and En-Hi SA with just 0.18% of mBERT model parameters (178M). There is a performance gap between the fine-tuning and PRO-CS approach on the NER dataset. We believe it is because the NER corpus is highly imbalanced across label classes. For example En-Hi NER corpus consists of 7 labels where 4 labels combined have less than 5% of total examples. Due to this skewed distribution of labels and limited model capacity for PRO-CS or PT, there is a drop in performance compared to FT. We expect this drop to reduce

[3] The same implemented by the transformers library https://huggingface.co



(a) Cross-Lingual transfer from En-Hi POS to En-Bn POS



(b) Cross-Task transfer from En-Hi SA to En-Hi HS

Figure 4: Comparison of PRO-CS with FT and PT with an increasing number of examples

as we increase model capacity through number of prompt tokens. Also, for En-Es NER, we observed a lot less code switching in the dataset, which also contributed to the performance drop.

6.1 Cross-lingual and cross-task transfer

We also compare the models in low-resource cross lingual and cross-task transfer settings. For cross-lingual transfer, we make use of En-Hi models to transfer to En-Bn (Bengali) and En-Ta (Tamil) due to the similarity between Indian languages. From Table 2, we observe that our approach performs better than prompt-tuning for both the languages. Furthermore, our model either exceeds or remains at par with the fine-tuned mBERT model. In the cross-task setup, we make use of the En-Hi sentiment analysis (SA) models for English-Hindi hate speech (HS) and intent classification (IN) task. As shown in Table 2, we outperform the prompt-tuning approach and achieve commensurate performance to the fine-tuning approach. These results clearly show our method's generalizability to other low resource CS language pairs.

7 Analysis

In this section, we perform a detailed analysis of our model's performance with increasing number of examples for cross-lingual and cross-task transfer. Further, we conduct an ablation study to show the effectiveness of each component in our model.

7.1 PRO-CS performance with increasing number of examples

We show PRO-CS performance comparison with prompt-tuning (PT) and fine-tuning (FT) with an increase in the number of training samples in both cross-lingual and cross-task settings. We experimented with total training instances ranging from 16 to 512. As shown in Figure 4, for cross-lingual transfer, PRO-CS is significantly more data efficient compared to both the fine-tuning and prompt-tuning approaches. Further, our proposed approach, PRO-CS, even outperforms the fine-tuning approach when the training data size is very small in both cross-lingual and cross-task settings. With increasing number of examples, the performance of PRO-CS matches that of FT with just 0.18% of model parameters.

7.2 Ablation study

Setting	En-Hi POS	En-Es POS
PRO-CS	60.33	82.98
w/o target lang prompt init	56.09	78.06
target lang prompt (frozen)	55.73	78.07
Shared subnetworks $\mathcal{G}$	60.33	77.47
Lang prompts only	56.67	77.75
Task prompts only	55.91	77.92

Table 3: Ablation study of the PRO-CS model on the POS task

We evaluate the contribution of each module of our proposed PRO-CS model by conducting a thorough ablation study. We ablate PRO-CS with different configurations (a) target language prompt not getting initialized from LID prompt; (b) non-trainable target language prompts; (c) shared attention subnetworks $\mathcal{G}$ to generate attention weights over language and task prompts; (d) only using language prompts; and (e) only using task prompts. We conduct these ablation studies on two language pairs (En-Hi, En-Es) for the POS task. Table 3 shows that each of our components contribute to

the performance of the model. Randomly initializing target language prompt shows a drop of 4 points, indicating the importance of LID prompt based initialization which gives the model an idea about code-switching. Additionally, having a different trainable language prompt for the target is also equally essential. We also observed that it is important to have separate attention subnetworks for language and task prompts to capture the contribution of language and task more precisely. The significance of task and language prompts is further supported by the amount of performance drop when we remove either of them.

8 Limitation

Datasets in code-switching are usually very unbalanced making the task harder. Our improvements in low-resource setting show that with a limited number of good examples, prompt-composition methods can achieve good results. Our models suffer when there is a skewed distribution of data. We also restrict our models to only classification tasks as we find that they are more ubiquitous in the CS domain. With the recent rise in translation between English and CS data, it will be interesting to apply our techniques for generation tasks like translation, open domain question answering, dialogue, etc. We have also made an assumption of existing monolingual corpora for training language-based prompts, which may not be true for various low-resource languages where code switching is prominent.

9 Conclusion

In this paper, we perform a detailed analysis of prompt-tuning techniques to show that they are effective in code-switching settings as well. However we find a significant gap between finetuning and widely used prompt-tuning. We address this gap by proposing a novel technique of prompt-composition PRO-CS for code-switching tasks. Our approach outperforms prompt-tuning technique across all 6 datasets in full-data setting with an average improvement of 3 and 3.5 points on sequence tagging and classification tasks respectively. This shows that composing prompts from source task and language prompt is more effective than training target-task only prompts. It also achieves competitive results to fine-tuning, even in low-resource cross-lingual and cross-task setting for both classification and sequence tagging tasks. For future work, we want to investigate these compositions for encoder-decoder T5 models for generation-based tasks. It would also be interesting to see if multi-task training helps in the downstream code-switch prompt-composition.

10 Ethical Considerations

There are various forms of code-switching that exist between different Asian and European languages, for example, Turkish-German (Çetinoğlu, 2017), Modern Arabic-Egyptian Arabic Aguilar et al. (2020). In this work, we evaluate our models on code-switching language pairs that are widely studied and used in the real world like En-Es, En-Hi. Our work does not undermine the existence of other code-switching language pairs. Instead, by showing effective transfer to other CS tasks, we aim to create language technologies to support the rise of CS scenarios on social media platforms and enable multilingual speakers to express their ideas with precision. We also believe that our work will present new directions for future research in the CS setting.

11 Acknowledgement

This work is partially funded by the Air Force Research Laboratory under agreement number FA8750-19-2-0200. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government. We are very grateful to Sanket Vaibhav Mehta for his insightful discussion and helpful feedback on our work.

References

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A Centralized Benchmark for Linguistic Code-switching Evaluation. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

Gustavo Aguilar and Thamar Solorio. 2020. From english to code-switching: Transfer learning with strong morphological clues. In *ACL*.

Fahad AlGhamdi, Giovanni Molina, Mona Diab, Thamar Solorio, Abdelati Hawwari, Victor Soto, and

Julia Hirschberg. 2016. Part of speech tagging for code switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 98–107, Austin, Texas. Association for Computational Linguistics.

Alan Ansell, Edoardo Maria Ponti, Anna Korhonen, and Ivan Vulić. 2021a. Composable sparse fine-tuning for cross-lingual transfer.

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021b. MAD-G: Multilingual adapter generation for efficient cross-lingual transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4762–4781, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Akari Asai, Mohammadreza Salehi, Matthew E. Peters, and Hannaneh Hajishirzi. 2022. Attentional mixtures of soft prompt tuning for parameter-efficient multi-task knowledge sharing.

Suman Banerjee, Nikita Moghe, Siddharth Arora, and Mitesh M. Khapra. 2018. A dataset for building code-mixed goal oriented conversation systems. In *COLING*.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Elad Ben-Zaken, Shauli Ravfogel, and Yoav Goldberg. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *ACL*.

Riyaz Ahmad Bhat, Rajesh Bhatt, Annahita Farudi, Prescott Klassen, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Ashwini Vaidya, Sri Ramagurumurthy Vishnu, et al. The hindi/urdu treebank project. In *Handbook of Linguistic Annotation*. Springer Press.

Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A dataset of Hindi-English code-mixed social media text for hate speech detection. In *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, pages 36–41, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Özlem Çetinoğlu. 2017. A code-switching corpus of Turkish-German conversations. In *Proceedings of the 11th Linguistic Annotation Workshop*, pages 34–40, Valencia, Spain. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020. Corpus creation for sentiment analysis in code-mixed Tamil-English text. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 202–210, Marseille, France. European Language Resources association.

Khyathi Chandu, Thomas Manzini, Sumeet Singh, and Alan W. Black. 2018. Language informed modeling of code-switched text. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 92–97, Melbourne, Australia. Association for Computational Linguistics.

Parul Chopra, Sai Krishna Rallabandi, Alan W Black, and Khyathi Raghavi Chandu. 2021. Switch point biased self-training: Re-purposing pretrained models for code-switching. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4389–4397, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *ArXiv*, abs/2003.10555.

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

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

Jinlan Fu, See-Kiong Ng, and Pengfei Liu. 2022. Polyglot prompt: Multilingual multitask promptraining. *ArXiv*, abs/2204.14264.

Demi Guo, Alexander Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. 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 4884–4896, Online. Association for Computational Linguistics.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *ICML*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

Anupam Jamatia, Björn Gambäck, and Amitava Das. 2016. Collecting and annotating indian social media code-mixed corpora. In *CICLing*.

Navya Jose, Bharathi Raja Chakravarthi, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. 2020. A survey of current datasets for code-switching research. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 136–141.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. Gluecos : An evaluation benchmark for code-switched NLP. *CoRR*, abs/2004.12376.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *CoRR*, abs/2104.08691.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *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)*, abs/2101.00190.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *CoRR*, abs/2110.07602.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Rudra Murthy, Pallab Bhattacharjee, Rahul Sharnagat, Jyotsana Khatri, Diptesh Kanojia, and Pushpak Bhattacharyya. 2022. Hiner: A large hindi named entity recognition dataset.

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A code mixed Hindi-English dataset and BERT language models. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 7–12, Marseille, France. European Language Resources Association.

Edward Ombui, Lawrence Muchemi, and Peter Wagacha. 2019. Hate speech detection in code-switched text messages. In *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–6.

Martha Palmer, Rajesh Bhatt, Bhuvana Narasimhan, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. Hindi syntax: Annotating dependency, lexical predicate-argument structure, and phrase structure. In *The 7th International Conference on Natural Language Processing*, pages 14–17.

Rana D. Parshad, Suman Bhowmick, Vineeta Chand, Nitu Kumari, and Neha Sinha. 2016. What is india speaking? exploring the "hinglish" invasion. *Physica A: Statistical Mechanics and its Applications*, 449:375–389.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Jung Pandey, Srinivas Pykl, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. *ArXiv*, abs/2008.04277.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020a. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulic, Iryna Gurevych, and Sebastian Ruder. 2020b. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. In *EMNLP*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Kumar Ravi and Vadlamani Ravi. 2016. Sentiment classification of hinglish text. In *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, pages 641–645.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *CoRR*, cs.CL/0306050.

Sebastin Santy, Anirudh Srinivasan, and Monojit Choudhury. 2021. BERTologiCoMix: How does code-mixing interact with multilingual BERT? In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 111–121, Kyiv, Ukraine. Association for Computational Linguistics.

Ayan Sengupta, Tharun Suresh, Md. Shad Akhtar, and Tanmoy Chakraborty. 2022. A comprehensive understanding of code-mixed language semantics using hierarchical transformer. *ArXiv*, abs/2204.12753.

Pranaydeep Singh and Els Lefever. 2020. Sentiment analysis for Hinglish code-mixed tweets by means of cross-lingual word embeddings. In *Proceedings of the The 4th Workshop on Computational Approaches to Code Switching*, pages 45–51, Marseille, France. European Language Resources Association.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W. Black. 2019. A survey of code-switched speech and language processing. *ArXiv*, abs/1904.00784.

Mariona Taulé, M. Antònia Martí, and Marta Recasens. 2008. AnCora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Shagun Uppal, Vivek Gupta, Avinash Swaminathan, Haimin Zhang, Debanjan Mahata, Rakesh Gosangi, Rajiv Ratn Shah, and Amanda Stent. 2020. Two-step classification using recasted data for low resource settings. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 706–719, Suzhou, China. Association for Computational Linguistics.

A. Ustun, Arianna Bisazza, Gosse Bouma, Gertjan van Noord, and Sebastian Ruder. 2022. Hyper-x: A unified hypernetwork for multi-task multilingual transfer. *ArXiv*, abs/2205.12148.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *CoRR*, abs/1905.00537.

Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2021. Are multilingual models effective in code-switching? *ArXiv*, abs/2103.13309.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Mengjie Zhao and Hinrich Schütze. 2021. Discrete and soft prompting for multilingual models. *ArXiv*, abs/2109.03630.

12 Appendix

12.1 Datasets

Table 4 mentions the sources of different source task datasets that we have used in our experiments. For Spanish, we translated the English-sentiment-imdb dataset, since we were not able to find any publicly available Spanish sentiment analysis dataset.

For code-switching datasets, we pick different types of pair of languages for different tasks. We choose the En-Hi sentiment analysis dataset from (Patwa et al., 2020), and the En-Es dataset from (Aguilar et al., 2020). en-ta was collected from (Chakravarthi et al., 2020) respectively. En-Hi POS and En-Es POS were collected from (Jamatia et al., 2016) nad (AlGhamdi et al., 2016) respectively. The NER for En-Hi and En-Es were taken from (Aguilar et al., 2020). We also use En-Bn POS dataset from ICON 2016 workshop [4]. We use the intent classification corpus for En-Hi from (Banerjee et al., 2018). The En-Hi hate speech dataset was taken from (Bohra et al., 2018).

Corpus	Type	Lang	#Train	#Dev
en-sentiment-imdb (Maas et al., 2011)		En	45k	5k
es-sentiment-imdb (translated)	SA	Es	45k	5k
hi-sentiment (Uppal et al., 2020)		Hi	4.3k	0.5k
wsj-pos (Marcus et al., 1993)		En	39.8k	1.3k
ud-spanish pos (Taulé et al., 2008)	POS	Es	14.2k	1.6k
ud-hindi-pos (Bhat et al.; Palmer et al., 2009)		Hi	13.3k	1.6k
conll-2003-ner (Sang and Meulder, 2003)		En	14.9k	3.4k
conll-2002-ner-es (Tjong Kim Sang, 2002)	NER	Es	8.3k	1.9k
hi-ner-2022 (Murthy et al., 2022)		Hi	76k	10k
hate-eval-2019 (Basile et al., 2019)	Hate	En	9k	1k

Table 4: Monolingual datasets of source tasks. The Spanish sentiment data set is created by translating the English IMDB data set.

[4] http://amitavadas.com/Code-Mixing.html