arXiv:2004.13240v2 [cs.CL] 18 Jun 2020

MultiMix: A Robust Data Augmentation Framework for Cross-Lingual NLP

M Saiful Bari *¶, **Tasnim Mohiuddin** *¶, **and Shafiq Joty**¶§
¶ Nanyang Technological University, Singapore
§ Salesforce Research Asia, Singapore
{bari0001@e., mohi0004@e., srjoty@}ntu.edu.sg

Abstract

Transfer learning has yielded state-of-the-art (SoTA) results in many supervised natural language processing tasks. However, annotated data for every target task in every target language is rare, especially for low-resource languages. We propose MultiMix, a novel data augmentation framework for self-supervised learning in zero-resource transfer learning scenarios. In particular, MultiMix targets to solve cross-lingual adaptation problems from a source language distribution to an unknown target language distribution, assuming no training labels are available for the target language task. At its core, MultiMix performs simultaneous self-training with data augmentation and unsupervised sample selection. To show its effectiveness, we conduct extensive experiments on zero-resource cross-lingual transfer tasks for Named Entity Recognition and Natural Language Inference. MultiMix achieves SoTA results in both tasks, outperforming the baselines by a good margin. With an in-depth model dissection, we demonstrate the cumulative contributions of different components to MultiMix's success.

1 Introduction

Self-supervised learning in the form of pretrained language models (LM) has been the driving force in developing state-of-the-art natural language processing (NLP) systems in recent years. These methods typically follow two basic steps, where a *supervised* task-specific fine-tuning follows a large-scale LM pretraining [12, 23, 25, 34, 49]. However, getting annotated data for every target task in every target language is difficult, especially when the target language is a low-resource one.

Recently, the *pretrain-finetune* paradigm has also been extended to multi-lingual setups to learn effective multi-lingual pretrained LMs that can be used for *zero-shot* cross-lingual transfer. Jointly trained deep contextualized multi-lingual LMs like mBERT [12], XLM [22], and XLM-R [11] coupled with supervised fine-tuning in the source language have been quite successful in transferring linguistic and task knowledge from one language to another without using any task labels in the target language. The joint pretraining with multiple languages allows these models to generalize across languages.

Despite the effectiveness of the pretrained multilingual LMs, several recent studies [20, 32] have also highlighted one crucial limiting factor for successful cross-lingual transfer. In particular, they all agree that the cross-lingual generalization ability of the model is limited by the (lack of) structural similarity between the source and target languages. For example, K et al. [20] report huge drops in zero-shot transferability when the two languages are structurally dissimilar. Specifically, for transferring from English, they report about 23.6% accuracy drop in Hindi (structurally dissimilar) compared to 9% drop in Spanish (structurally similar) in the cross-lingual natural language inference (XNLI) task.

*equal contribution

Preprint. Under review.

One attractive way to improve cross-lingual generalization is to perform *data augmentation* [39], and train the pretrained LM (*e.g.,* XLM-R) on examples that are similar but different from the labeled data in the source language. Formalized by the Vicinal Risk Minimization (VRM) principle [8], such data augmentation methods have shown impressive results recently in computer vision [7, 24, 51]. These methods enlarge the support of the training distribution by generating *new* data points from a *vicinity distribution* around each training example. For images, the vicinity of a training image can be defined by a set of operations like rotation and scaling, or by linear mixtures of features and labels [51]. However, when it comes to text, such augmentation methods have rarely been successful. The main reason is that unlike images, linguistic units (*e.g.,* words, phrases) are discrete and a smooth change in their embeddings may not result in a plausible linguistic unit that has similar meanings.

In NLP, the most successful data augmentation method has so far been *back-translation* [38] which generates paraphrases of an input sentence through round-trip translations. However, back-translation requires parallel data to train effective machine translation systems, acquiring which can be more expensive for low-resource languages than annotating the target language data with task labels. Furthermore, back-translation is only applicable in a supervised setup and to tasks where it is possible to find the alignments between the original labeled entities and the back-translated entities, such as in question answering [13, 50]. It is also not obvious how to apply back-translation to augment labeled data for sequence tagging tasks like named entity recognition (NER).

In this work, we propose MultiMix, a generic data augmentation framework for improving cross-lingual generalization of multilingual pretrained LMs. MultiMix augments data from the unlabeled training examples in the target language as well as from the virtual input samples (*e.g.,* sentences) generated from the vicinity distribution of the source and target language sentences. With the augmented data, MultiMix performs simultaneous *self-learning* with an effective distillation strategy to learn a strongly adapted cross-lingual model from noisy (pseudo) labels for the target language task. We propose novel ways to generate virtual input samples using a multilingual masked LM [11], and get reliable task labels by simultaneous multilingual co-training.

We validate the effectiveness and robustness of MultiMix by performing extensive experiments on two very different zero-resource cross-lingual transfer tasks – cross-lingual NER (XNER) and XNLI, which posit different sets of challenges. Our results demonstrate that MultiMix yields about 5.5% absolute improvement on average over the baseline for zero-resource XNER, setting a new SoTA for all the five languages. For XNLI, with only 5% labeled data in the source, MultiMix gets comparable results to the baseline that uses all the labeled data, and surpasses the standard baseline by 2.55% on average when it uses all the labeled data in the source. We provide a comprehensive analysis of the factors that contribute to MultiMix's performance. Our framework will be open-sourced.

2 Background

Contextual representation and cross-lingual transfer In recent years, significant progress has been made in learning contextual word representations and pretrained models [12, 17, 31, 34, 49]. Notably, BERT [12] pretrains a Transformer [43] encoder with a masked language model (MLM) objective, and uses the same model architecture to adapt to a new task. It also comes with a multilingual version mBERT, which is trained jointly on 102 languages. RoBERTa [25] extends BERT with improved training, while XLM [22] extends mBERT with a conditional LM and a translation LM (using parallel data) objectives. Conneau et al. [11] train the largest multilingual language model XLM-R with RoBERTa framework.

Despite any explicit cross-lingual supervision, mBERT and its variants have been shown to learn cross-lingual representations that generalize well across languages. Wu and Dredze [47] and Pires et al. [32] evaluate the zero-shot cross-lingual transferability of mBERT on several tasks and attribute its generalization capability to shared subword units. Pires et al. [32] also found structural similarity (*e.g.,* word order) to be another important factor for successful cross-lingual transfer. K et al. [20], however, show that the shared subword has a minimal contribution; instead, the structural similarity between languages is more crucial for effective transfer.

Vicinal risk minimization Data augmentation supported by the Vicinal Risk Minimization (VRM) principle [8] can be an effective choice for achieving better out-of-distribution adaptation. In VRM, we minimize the empirical vicinal risk defined as: $\mathcal{L}_v(\theta) = \frac{1}{N}\sum_{n=1}^{N} l(f_\theta(\tilde{x}_n), \tilde{y}_n)$, where f_θ denotes the model parameterized by θ, and $\mathcal{D}^{\text{aug}} = \{(\tilde{x}_n, \tilde{y}_n)\}_{n=1}^{N}$ is an augmented dataset constructed by

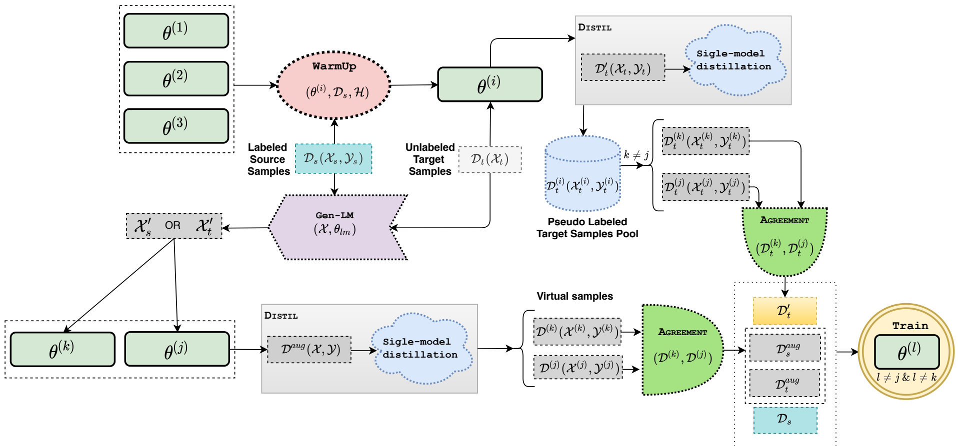


Figure 1: MultiMix framework. After training the base task models ($\theta^{(i)}$) on source labeled data $\mathcal{D}_s$ (WarmUp), we use two models to *distil* target language samples that are later used for training data augmentation for the third model. A pre-trained LM (GEN-LM) is used to generate new training samples ($\mathcal{X}_s^{'}$, $\mathcal{X}_t^{'}$) for both source and target languages which are then labeled with the corresponding $\theta^{(i)}$ task model. The generated samples with pseudo labels are then added to the training data for the target model by co-guessing (AGREEMENT) from the other two models.

sampling the vicinal distribution $\vartheta(\tilde{x}_i, \tilde{y}_i | x_i, y_i)$ around the original training sample (x_i, y_i). Defining vicinity is however quite challenging as it requires the extraction of samples from a distribution without hurting their labels. Earlier methods apply simple rules like rotation and scaling of images [39]. Recent work [7, 24, 51] show impressive results in image classification with simple linear interpolation of data. However, to our knowledge, none of these methods have so far been successful in NLP due to the discrete nature of texts.

3 MultiMix framework

While recent cross-lingual transfer learning efforts have relied almost exclusively on multi-lingual pre-training and zero-shot transfer of a fine-tuned source model, there is a great potential for more elaborate methods that can leverage the unlabeled data better. Motivated by this, we present MultiMix - our generic data augmentation framework for zero-resource cross-lingual task adaptation.

Figure 1 gives an overview of MultiMix. Let $\mathcal{D}_s = (\mathcal{X}_s, \mathcal{Y}_s)$ and $\mathcal{D}_t = (\mathcal{X}_t)$ denote the training data for a source language s and a target language t, respectively. MultiMix augments data from various origins at different stages of learning. In the initial stage, it uses the training examples in the target language, $\mathcal{X}_t$. In later stages, it uses the virtual input samples (i.e. sentences) generated from the vicinity distribution of source and target examples: $\vartheta(\tilde{x}_n^s | x_n^s)$ and $\vartheta(\tilde{x}_n^t | x_n^t)$, where $x_n^s \sim \mathcal{X}_s$ and $x_n^t \sim \mathcal{X}_t$. MultiMix performs *self-training* on the augmented data to acquire the corresponding (pseudo) labels. To avoid *confirmation bias* with self-training where the model accumulates its own errors, it simultaneously trains three "task" models (inspired by [52]) to generate *virtual* training data ($\mathcal{D}^{\text{aug}}$) through data augmentation and filtering of potential label noises via multi-epoch *co-teaching*. In each epoch, the co-teaching process first performs *co-distillation*, where two peer task models are used to select the unlabeled training examples from the target language for the third task model. Then the generated samples with pseudo labels are added to the target task model's training data by taking the agreement from the other two models, a process we refer to as *co-guessing*. In this way, in each epoch, each of the three task models is trained with the augmented source and/or target language data filtered by other two peer models. We present the training method of MultiMix in Algorithm 1.

In our case, each of the *task* models is an instance of XLM-R [11] fine-tuned on the task in the source language, whereas the pre-trained masked LM parameterized by θ_{lm} (i.e. before fine-tuning on the task) is used to define the *vicinity* distribution $\vartheta(\tilde{x}_n | x_n, \theta_{lm})$ around each selected example x_n.

3.1 Warm-up step: training task models with confidence penalty

In the initial step, we train three instances of the XLM-R model ($\theta^{(1)}, \theta^{(2)}, \theta^{(3)}$) with an additional task-specific linear layer on the source language (English) labeled data. Each model has the same

Algorithm 1 MultiMix: a robust data augmentation framework for cross-lingual NLP

Input: *source* (s) and *target* (t) language datasets: $\mathcal{D}_s = (\mathcal{X}_s, \mathcal{Y}_s), \mathcal{D}_t = (\mathcal{X}_t)$; task models: $\theta^{(1)}, \theta^{(2)}, \theta^{(3)}$, pre-trained masked LM θ_{lm}, mask ratio P, diversification factor δ, sampling factor α, and distillation factor η
Output: models trained on augmented data

$\theta^{(1)}, \theta^{(2)}, \theta^{(3)} = \textsc{WarmUp}(\mathcal{D}_s = (\mathcal{X}_s, \mathcal{Y}_s), \theta^{(1)}, \theta^{(2)}, \theta^{(3)})$ ▷ warm up with conf. penalty.
for $e \in [1:3]$ **do** ▷ e is for epoch.
  for $k \in \{1, 2, 3\}$ **do**
    $\mathcal{X}_t^{(k)}, \mathcal{Y}_t^{(k)} = \textsc{Distil}(\mathcal{X}_t, \eta_e, \theta^{(k)})$ ▷ infer and select tgt training data for augmentation.
    for $j \in \{1, 2, 3\}$ **do**
      if $k == j$ **then** Continue
      `/* source language data augmentation */`
      $\mathcal{X}_s' = \textsc{Gen-LM}(\mathcal{X}_s, \theta_{lm}, P, \delta)$
      $\mathcal{X}_s^{(k)}, \mathcal{Y}_s^{(k)} = \textsc{Distil}(\mathcal{X}_s', \eta_e, \theta^{(k)})$; $\mathcal{X}_s^{(j)}, \mathcal{Y}_s^{(j)} = \textsc{Distil}(\mathcal{X}_s', \eta_e, \theta^{(j)})$
      $\mathcal{D}_s^{\text{aug}} = \textsc{Agreement}\big(\mathcal{D}_s^{(k)} = (\mathcal{X}_s^{(k)}, \mathcal{Y}_s^{(k)}), \mathcal{D}_s^{(j)} = (\mathcal{X}_s^{(j)}, \mathcal{Y}_s^{(j)})\big)$
      `/* target language data selection */`
      $\mathcal{X}_t^{(j)}, \mathcal{Y}_t^{(j)} = \textsc{Distil}(\mathcal{X}_t, \eta_e, \theta^{(j)})$
      $\mathcal{D}_t' = \textsc{Agreement}\big(\mathcal{D}_t^{(k)} = (\mathcal{X}_t^{(k)}, \mathcal{Y}_t^{(k)}), \mathcal{D}_t^{(j)} = (\mathcal{X}_t^{(j)}, \mathcal{Y}_t^{(j)})\big)$
      `/* target language data augmentation */`
      $\mathcal{X}_t' = \textsc{Gen-LM}(\mathcal{X}_t, \theta_{lm}, P, \delta)$
      $\mathcal{X}_t^{(k)}, \mathcal{Y}_t^{(k)} = \textsc{Distil}(\mathcal{X}_t', \eta_e, \theta^{(k)})$; $\mathcal{X}_t^{(j)}, \mathcal{Y}_t^{(j)} = \textsc{Distil}(\mathcal{X}_t', \eta_e, \theta^{(j)})$
      $\mathcal{D}_t^{\text{aug}} = \textsc{Agreement}\big(\mathcal{D}_t^{(k)} = (\mathcal{X}_t^{(k)}, \mathcal{Y}_t^{(k)}), \mathcal{D}_t^{(j)} = (\mathcal{X}_t^{(j)}, \mathcal{Y}_t^{(j)})\big)$
      `/* train new models on augmented data */`
      for $l \in \{1, 2, 3\}$ **do**
        if $l \neq j$ and $l \neq k$ **then**
          with sampling factor α,
          train $\theta^{(l)}$ with $\mathcal{D} = \{\mathcal{D}_s \mathbf{1}_{\{1,3\}}(e) \cup \mathcal{D}_t' \mathbf{1}_{\{1,3\}}(e) \cup \mathcal{D}_s^{\text{aug}} \mathbf{1}_{\{3\}}(e) \cup \mathcal{D}_t^{\text{aug}} \mathbf{1}_{\{2,3\}}(e)\}$
Return $\{\theta^{(1)}, \theta^{(2)}, \theta^{(3)}\}$

architecture (XLM-R large) but is initialized with different random seeds. For token-level prediction tasks (*e.g.,* NER), the *token-level* representations are fed into the classification layer, whereas for sentence-level tasks (*e.g.,* XNLI) the `[CLS]` representation is used as input to the classification layer. For each task, we fine-tune all the model parameters end-to-end.

Confidence penalty We train the task models using the cross-entropy (CE) loss. However, maximum likelihood training with the standard CE loss may result in overfitted models that produce overly confident predictions (low entropy), especially when the class distribution is not balanced. This may impose difficulties on our sample selection methods in isolating good samples from noisy ones [24]. Following [30], we address this issue by adding a negative entropy term $-\mathcal{H}$ to the CE loss.

$$\mathcal{L}(\theta) = -\sum_{c=1}^{C} \underbrace{y^c \log p_\theta^c(\mathbf{x})}_{\text{CE}} + \underbrace{p_\theta^c(\mathbf{x}) \log p_\theta^c(\mathbf{x})}_{-\mathcal{H}} \tag{1}$$

where $\mathbf{x}$ is the representation from XLM-R that goes to the output layer, and y_n^c and $p_\theta^c(\mathbf{x}_n)$ are respectively the ground truth label and model predictions with respect to class c.

3.2 Vicinity distribution and sentence augmentation

Our augmentated data comes from two different sources: the *original* target language samples $\mathcal{X}_t$, and the *virtual* samples generated from the vicinity distribution of the source and target examples: $\vartheta(\tilde{x}_n^s|x_n^s, \theta_{lm})$ and $\vartheta(\tilde{x}_n^t|x_n^t, \theta_{lm})$, where $x_n^s \sim \mathcal{X}_s$ and $x_n^t \sim \mathcal{X}_t$. It has been shown that contextual LMs pre-trained on large-scale datasets capture useful linguistic features and can be used to generate fluent grammatical texts [16]. We use the XLM-R masked LM [11] as our vicinity model θ_{lm}, which is trained on massive multilingual corpora (2.5 TB of Common-Crawl data in 100 languages). Note that the vicinity model is a disjoint pre-trained entity whose weights are not trained on any task objective.

In order to generate samples around each *selected* example, we first randomly choose $P\%$ of the input tokens. Then we successively (*i.e.*, one at a time) mask one of the chosen tokens and ask θ_{lm} to predict a token in that masked position, *i.e.*, we compute $\vartheta(\tilde{x}_m|x, \theta_{lm})$ with m being the index of the masked token. For a specific mask, we sample S candidate words from the output distribution. We then generate novel sentences by following one of the two alternative approaches.

- **Successive max** In this approach, we take the most probable output token ($S = 1$) at each prediction step, $\hat{o}_m = \arg\max_o \vartheta(\tilde{x}_m = o|x, \theta_{lm})$. A new sentence is then constructed by $P\%$ newly generated tokens. We generate δ virtual samples for each original example x, by randomly masking $P\%$ tokens each time. Here, δ is the diversification factor.
- **Successive cross** In this approach, we divide each original sample x into two parts and use successive max to create two sets of augmented samples of size δ_1 and δ_2 respectively. We then take the cross of these two sets to generate $\delta_1 \times \delta_2$ augmented samples.

Augmentation of sentences through *successive max or cross* is carried out within the GEN-LM (generate via LM) module in Algorithm 1.

3.3 Co-labeling of augmented sentences through distillation

Traditional VRM based data augmentation methods assume that the samples generated by the vicinity model share the same class so that the same class labels can be used for the newly generated data [8]. This approach does not consider the vicinity relation across examples of different classes. Recent methods in computer vision [7, 24, 51] relax this assumption and generate new images and their labels as simple *linear interpolations*. However, as mentioned earlier, due to the discrete nature of texts, such linear interpolation methods have not been successful so far in NLP. The meaning of a sentence (*e.g.,* sentiment, word meanings) can change entirely even with minor variations in the original sentence. For example, consider the following example generated by our vicinity model.

Original text:	*EU rejects German call to boycott british lamb.*
Masked text:	*<mask> rejects german call to boycott british lamb.*
MLM prediction:	*Trump rejects german call to boycott british lamb.*

Here, EU is an *Organization* whereas the newly predicted word *Trump* is a *Person* (different named entity type). Therefore, we need to relabel the augmented sentences no matter whether the original sentence has labels (source language sentences) or not (target language sentences).

However, the relabeling process can induce noise, especially for the target language sentences, since the task model may not be adapted fully in the early stages of training. We propose two stages of distillation to filter out noisy augmented samples, as we describe below.

Single-model distillation. The first stage of distillation involves predictions from a single peer model for which we propose two alternatives:

(*i*) *Distillation by model confidence:* In this approach, we select samples based on the model's prediction confidence. This method is similar in spirit with the selection method proposed in [35]. For sentence-level tasks (*e.g.,* XNLI), the model produces a single class distribution for each training example. In this case, the model's confidence is computed by $\hat{p} = \max_{c\in\{1...C\}} p^c_\theta(\mathbf{x})$. For token-level sequence labeling tasks (*e.g.,* NER), the model's confidence is computed by:

$$\hat{p} = \frac{1}{T}\Big\{ \max_{c\in\{1...C\}} p^c_\theta(\mathbf{x}_t)\Big\}_{t=1}^{T} \tag{2}$$

where T is the length of the sequence. The distillation is then done by selecting the top $\eta\%$ samples with the highest confidence scores.

(*ii*) *Distillation by clustering:* Here our goal is to cluster the samples based on their *goodness*. It has been shown in computer vision that deep models tend to learn good samples faster than noisy ones, leading to a lower loss for good samples and higher loss for noisy ones [2, 15]. Inspired by [1, 24], we propose to model *per-sample loss distribution* with a mixture model, which we fit using an *Expectation-Maximization* (EM) algorithm. However, contrary to those approaches which use actual (supervised) labels, we use the model predicted pseudo labels to compute the loss for the samples. We use a 1d two-component Gaussian Mixture Model (GMM) due to its flexibility in modeling the sharpness of a distribution [24]. For each sample $\mathbf{x}$, its *goodness* probability is the posterior probability $p(z = g|\mathbf{x}, \theta_{\text{GMM}})$, where g is the component with smaller mean loss. Here, distillation hyperparameter η is the posterior probability threshold based on which samples are selected.

Distillation by model agreement. In the second stage of distillation, we select samples by taking the agreement (co-guess) of two different peer models $\theta^{(j)}$ and $\theta^{(k)}$ to train the third $\theta^{(l)}$. Formally,

$$\text{AGREEMENT}\big(\mathcal{D}^{(k)}, \mathcal{D}^{(j)}\big) = \{(\mathcal{X}^{(k)}, \mathcal{Y}^{(k)}) : \mathcal{Y}^{(k)} = \mathcal{Y}^{(j)}\} \quad s.t.\ k \neq j$$

3.4 Data samples manipulation

MultiMix uses multi-epoch co-teaching. It uses $\mathcal{D}_s$ and $\mathcal{D}_t$ in the first epoch. In epoch 2, it uses $\mathcal{D}_t^{aug}$, and finally it uses all the four datasets - $\mathcal{D}_s$, $\mathcal{D}_t$, $\mathcal{D}_t^{aug}$, and $\mathcal{D}_s^{aug}$ (line 22 in Alg. 1). The datasets used at different stages can be of different sizes. For example, the number of augmented samples in $\mathcal{D}_s^{aug}$ and $\mathcal{D}_t^{aug}$ grow polynomially with the *successive cross* masking method. Also, the *co-distillation* produces sample sets of variable sizes. To ensure that our model does not overfit on one particular dataset, we employ a balanced sampling strategy. For N number of datasets $\{\mathcal{D}_i\}_{i=1}^N$ with probabilities, $\{p_i\}_{i=1}^N$, we define the following multinomial distribution to sample from:

$$p_i = \frac{f_i^\alpha}{\sum_{j=1}^N f_j^\alpha} \text{ where } f_i = \frac{n_i}{\sum_{j=1}^N n_j} \tag{3}$$

where α is the sampling factor and n_i is the total number of samples in the i^{th} dataset. By tweaking α, we can control how many samples a dataset can provide in the mix.

4 Experiments

We consider two cross-lingual tasks in the zero-resource transfer setting: cross-lingual NER (XNER) and NLI (XNLI). We assume labeled training data is available only in English, and transfer the trained model to a target language. As a token-level sequence labeling task, XNER evaluates the model's capability to learn task-specific contextual representations that depend on language structure. XNLI, on the other hand, judges the model's ability to extract a reasonable meaning representation of sentences across different languages [10]. For all of our experiments in XNER and XNLI, we report the average score of the three models (which uses different seeds) with standard deviation ($\pm$).

4.1 Datasets

XNER For XNER, we transfer from English (en) to Spanish (es), German (de), Dutch (nl), Arabic (ar), and Finnish (fi). For English and German, we consider the dataset from CoNLL-2003 shared task [37], while for Spanish and Dutch, we use the dataset from CoNLL-2002 shared task [36]. We collected the Arabic and Finnish NER datasets from Bari et al. [6].

XNLI We use the standard XNLI dataset [10] which extends the MultiNLI dataset [45] to 15 languages. For a given pair of sentences, the task is to predict the entailment relationship between the two sentences, *i.e.*, whether the second sentence (*hypothesis*) is an *Entailment*, *Contradiction*, or *Neutral* with respect to the first one (*premise*). For XNLI, we experiment with transferring from English to Spanish (es), German (de), Arabic (ar), Swahili (sw), Hindi (hi), and Urdu (ur).

4.2 Task and model settings

Our objective is to adapt a task model from a source (language) distribution to an unknown target (language) distribution assuming no labeled data in the target language. In this scenario, there might be two different distributional gaps: (*i*) the generalization gap for the source distribution, and (*ii*) the gap between the source and target language distribution. We wish to investigate our method in tasks that exhibit such properties. We use the standard task setting for XNER, where we take 100% samples from the datasets as they come from various domains and sizes without any specific bias.

However, the XNLI dataset comes with machine-translated texts in target languages. Thus, the data is parallel and arguably lacks enough diversity in domains (*i.e.,* source and target come from the same domain). Cross-lingual models trained in the standard setup may pick up distributional bias (in the label space) from the source. Artetxe et al. [5] also argue that the translation process can induce subtle artifacts that may have a notable impact on models. Therefore, for XNLI, we experiment with two different task setups. First, to ensure distributional differences and non-parallelism, we use 5% of the training data from the source language and augment a different (nonparallel) 5% dataset for the target language. We used a different seed each time to retrieve the 5% target language data. Second, to compare with previous methods, we also evaluate on the standard 100% setup. However, the evaluation is done on the entire testset in both setups. We will refer to these two settings as **XNLI-5%** and **XNLI-100%**. More details about model settings and hyperparameters are in the Appendix.

4.3 Results

XNER From Table 5, we observe that after performing *warm-up* step with conf-penalty ($\mathcal{H}$), XLM-R performs better than mBERT on average by $\sim$ 3.8% for all the languages. On average, MultiMix

Table 1: Results in **F1 score** for Cross-lingual Named Entity Recognition (XNER). "x" represents model fails to converge and "-" represents no results were reported for the setup.

Model	en	es	nl	de	ar	fi
			Supervised Result			
(Char+fastText) bi-LSTM-CRF [6]	89.77 ± 0.19	84.71 ± 0.06	85.16 ± 0.21	78.14 ± 0.42	75.49 ± 0.53	84.21 ± 0.13
XLM-R [11]	92.92	89.72	92.53	85.81	–	–
			Zero-Resource Baseline			
fastText-bi-LSTM-CRF [6]	88.98 ± 0.25	x	x	x	x	x
(Char+fastText)bi-LSTM-CRF [6]	89.92 ± 0.15	26.76 ± 1.45	20.94 ± 0.74	8.34 ± 1.43	x	22.44 ± 2.23
BERT-base-cased	91.21 ± 0.18	52.88 ± 1.33	29.16 ± 3.30	44.41 ± 2.36	x	30.18 ± 1.93
Wu and Dredze [47]	-	74.96	77.57	69.56	-	-
Pires et al. [32]	-	73.59	77.36	69.74	-	-
Conneau et al. [11]	-	78.64	80.80	71.40	-	-
Bari et al. [6]	-	75.93 ± 0.81	74.61 ± 1.24	65.24 ± 0.56	36.91 ± 2.74	53.77 ± 1.54
mBERT-cased (Our implementation)	91.13 ± 0.14	74.76 ± 1.06	79.58 ± 0.38	70.99 ± 1.24	45.48 ± 1.47	65.95 ± 0.76
XLM-R (Our implementation)	92.23 ± 0.19	79.29 ± 0.43	80.87 ± 0.90	73.40 ± 0.96	49.04 ± 1.19	75.57 ± 0.94
			Our Method			
mBERT-cased + conf-penalty	90.81 ± 0.17	75.06 ± 0.63	79.26 ± 0.65	72.31 ± 0.52	47.03 ± 1.65	66.72 ± 0.44
XLM-R + conf-penalty	92.49 ± 0.09	80.45 ± 0.42	81.07 ± 0.12	73.76 ± 1.01	49.94 ± 0.43	76.05 ± 0.25
MultiMix	–	**83.05 ± 0.38**	**85.21 ± 0.23**	**80.33 ± 0.07**	**57.35 ± 0.56**	**79.75 ± 0.34**

gives a sizable improvement of $\sim$5.5% on five different languages. Specifically, we get an absolute improvement of 3.76%, 4.34%, 6.94%, 8.31%, and 4.18% for *es, nl, de, ar,* and *fi*, respectively.

Interestingly, MultiMix surpasses the *supervised* bi-LSTM-CRF for *nl* and *de* without using any labeled data in the target language. It also produces comparable results for es with the supervised bi-LSTM-CRF baseline and a significant improvement for *ar* and *fi* over the zero-shot baselines.

XNLI-5% From Table 2, we see that the performance of the XLM-R model trained on 5% training data is surprisingly good compared to the model trained on the full data (XLM-R (Our implementation)), lagging by only 5.6% on average. In our single GPU implementation of XNLI, we could not reproduce the reported results of Conneau et al. [11]. However, our results resemble the reported XLM-R results of XTREME [18]. We consider XTREME as our standard baseline for XNLI-100%.

We observe that with only 5% labeled data in the source, Multimix gets comparable results to the XTREME baseline that uses 100% labeled data (lagging behind by only $\sim$ 0.7% on average); even for *ar* and *sw*, we get 0.22% and 1.11% improvements, respectively. It surpasses the standard 5% baseline by 4.2% on average. Specifically, MultiMix gets absolute improvements of 3.05%, 3.34%, 5.38%, 5.01%, 4.29%, and 4.12% for *es, de, ar, sw, hi,* and *ur*, respectively.

Table 2: Results in Accuracy for Cross-lingual Natural Language Inference (XNLI) task.

Model	en	es	de	ar	sw	hi	ur
			Supervised Result (TRANSLATE-TRAIN-ALL)				
Huang et al. (Wiki+MT) [19]	85.6	82.3	80.9	78.2	73.8	73.4	69.6
XLM-R (Base) [11]	85.4	82.2	80.3	77.3	73.1	76.1	73.0
XLM-R [11]	89.1	86.6	85.7	83.1	78.0	81.6	78.1
			Zero-Resource Baseline for **Full (100%) English labeled** training set				
mBERT-cased [47]	82.1	74.3	71.1	64.9	50.4	60.0	58.0
XLM [22]	83.2	76.3	74.2	68.5	64.6	65.7	63.4
XLM-R (Paper) [11]	89.1	85.1	83.9	79.8	73.9	76.9	73.8
XLM-R (XTREME) [18]	88.7	83.7	82.5	77.2	71.2	75.6	71.7
XLM-R (Our implementation)	88.87 ± 0.31	84.34 ± 0.37	82.78 ± 0.56	78.44 ± 0.50	72.08 ± 1.05	76.40 ± 0.87	72.10 ± 1.22
			Our Method				
XLM-R + conf-penalty	88.83 ± 0.12	84.30 ± 0.24	82.86 ± 0.14	78.20 ± 0.38	71.83 ± 0.41	76.24 ± 0.47	71.62 ± 0.70
MultiMix	–	**85.65 ± 0.04**	**84.18 ± 0.46**	**80.50 ± 0.19**	**74.70 ± 0.47**	**78.84 ± 0.32**	**73.35 ± 0.41**
			Zero-Resource Baseline for **5% English labeled** training set				
XLM-R (Our implementation)	83.08 ± 1.04	78.48 ± 0.76	77.54 ± 0.60	72.04 ± 0.79	67.3 ± 0.66	70.41 ± 0.09	66.72 ± 0.29
			Our Method				
XLM-R + conf-penalty	84.24 ± 0.22	79.23 ± 0.37	78.47 ± 0.20	72.43 ± 0.75	67.72 ± 0.17	71.08 ± 0.73	67.63 ± 0.62
MultiMix	–	**81.53 ± 0.11**	**80.88 ± 0.28**	**77.42 ± 0.15**	**72.31 ± 0.12**	**74.70 ± 0.26**	**70.84 ± 0.22**

XNLI-100% Now, considering MultiMix's performance on the full (100%) labeled source data in Table 2, we see that it achieves state-of-the-art results for 5 out of 6 languages with an absolute improvement of 2.55% on average from the XTREME baseline. Specifically, MultiMix gets absolute improvements of 1.95%, 1.68%, 4.30%, 3.50%, 3.24%, and 1.65% for *es, de, ar, sw, hi,* and *ur*, respectively.

5 MultiMix framework analysis

In this section, we further analyze our MultiMix framework by dissecting it and measuring the contribution of its different components.

5.1 Analysis of distillation methods

Model confidence vs. clustering We first analyze the performance of our *single-model distillation* methods (§3.3) to see which of the two alternatives works better. From Table 3, we see that both perform similarly, with *model confidence* being slightly better. In our main experiments (Tables 5-2) and subsequent analysis, we use model confidence for distillation. However, we should not rule out the clustering method as it gives a more general solution to consider other distillation features (*e.g.,* sequence length, language) than model prediction scores, which we did not explore in this paper.

Distillation factor η We next show the results for different distillation factor (η) in Table 3. Here 100% refers to the case when no single-model distillation is done based on model confidence. We notice that the best results for each of the languages are obtained for values other than 100%, which indicates that distillation is indeed an effective step in MultiMix. See Appendix for more on η.

Two-stage distillation We now validate whether the second stage distillation (*distillation by model agreement*) is needed. In Table 3, we also compare the results with the model agreement (shown as $\cap$) to the results without using any agreement (shown as ϕ). We observe better performance with *model agreement* in all the cases on top of the single-model distillation.

Table 3: Analysis of **distillation** on XNER. Results after epoch-1 training that uses $\{\mathcal{D}_s, \mathcal{D}_t\}$.

η	Agreement	es	nl	de	ar	fi
Distillation by clustering						
.7	$\cap$	82.28	83.25	78.86	52.64	78.47
.5	$\cap$	82.35	83.11	78.16	54.20	78.28
Distillation by model confidence						
50%	$\cap$	**82.52**	82.46	75.95	52.00	77.51
	ϕ	81.66	82.26	77.19	52.97	77.77
80%	$\cap$	82.33	**83.53**	78.50	**54.48**	78.43
	ϕ	81.61	83.03	77.08	53.31	78.34
90%	$\cap$	81.90	82.80	**79.03**	52.41	**78.66**
	ϕ	81.21	82.77	77.28	52.20	77.93
100%	$\cap$	82.50	82.35	77.06	52.58	77.51
	ϕ	81.89	82.15	76.97	52.68	78.01

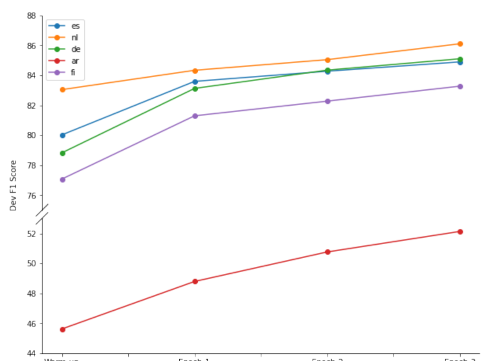


Figure 2: Validation F1 results in XNER for multi-epoch co-teaching training of MultiMix.

5.2 Different types of augmentation in different stages

Figure 2 presents the effect of different types of augmented data in our multi-epoch co-teaching framework. We observe that in every epoch of co-teaching, there is a significant boost in F1 scores for each of the languages on the XNER task. Compared to other languages, zero-shot transfer for Arabic is low. We suspect this is because of the structural dissimilarity between Arabic and English [20]. However, the improvement for Arabic is consistent with (and even higher than) other languages.

6 Conclusion

We propose a novel data augmentation framework, MultiMix, for zero-resource cross-lingual task adaptation. Our method performs simultaneous self-training with data augmentation and unsupervised sample selection. MultiMix augments data from the unlabeled training examples in the target language as well as from the vicinity distribution of the source and target language samples.

With extensive experiments of two cross-lingual tasks of different natures, we have demonstrated the effectiveness of MultiMix. For the zero-resource XNER task, MultiMix sets a new SoTA for all the five languages. For the XNLI task, with only 5% labeled data in the source, MultiMix gets comparable results to the baseline that uses 100% labeled data. Through an in-depth analysis, we show the cumulative contributions of different components of MultiMix.

Broader Impact

Our work has a potential impact on the neural cross-lingual task adaptation. This will enable the language model to adapt better on the low-resource as well as high-resource languages. It will foster the research of low resource languages in a new direction, which will enable the content creator to create content for a broader audience. It may also help them prevent online bullying and detect fake news in multiple languages. We hope MultiMix will become a generic framework for cross-lingual task adaptation. Another positive impact to the society is that our method can make NLP more inclusive of languages (and their speakers) for which there are not enough resources. In this paper, we have experimented with *Finnish, Arabic, Hindi, Urdu, Swahili*, in addition to *English, Spanish,* and *Dutch*, and show that our methods generalize well across all the languages. On the negative side, it requires GPU compute, which is not good for the environment/climate.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning (ICML)*. June 2019.

[2] Devansh Arpit, Stanisław Jastrzundefinedbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 233–242. JMLR.org, 2017.

[3] Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *CoRR*, abs/1812.10464, 2018. URL `http://arxiv.org/abs/1812.10464`.

[4] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations, 2019.

[5] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Translation artifacts in cross-lingual transfer learning, 2020.

[6] M Saiful Bari, Shafiq Joty, and Prathyusha Jwalapuram. Zero-Resource Cross-Lingual Named Entity Recognition. In *Proceedings of the 34th AAAI Conference on Artifical Intelligence*, AAAI '20, pages xx–xx, New York, USA, 2020. AAAI.

[7] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5050–5060. Curran Associates, Inc., 2019.

[8] Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 416–422. MIT Press, 2001. URL `http://papers.nips.cc/paper/1876-vicinal-risk-minimization.pdf`.

[9] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *CoRR*, abs/1710.04087, 2017. URL `http://arxiv.org/abs/1710.04087`.

[10] Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: evaluating cross-lingual sentence representations. *CoRR*, abs/1809.05053, 2018. URL `http://arxiv.org/abs/1809.05053`.

[11] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. 2019.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://www.aclweb.org/anthology/N19-1423`.

[13] Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1091. URL `https://www.aclweb.org/anthology/D17-1091`.

[14] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[15] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8527–8537. Curran Associates, Inc., 2018.

[16] John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL `https://www.aclweb.org/anthology/N19-1419`.

[17] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL `https://www.aclweb.org/anthology/P18-1031`.

[18] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *CoRR*, abs/2003.11080, 2020.

[19] Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2485–2494, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1252. URL `https://www.aclweb.org/anthology/D19-1252`.

[20] Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. Cross-lingual ability of multilingual {bert}: An empirical study. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HJeT3yrtDr`.

[21] Phillip Keung, yichao lu, and Vikas Bhardwaj. Adversarial learning with contextual embeddings for zero-resource cross-lingual classification and ner. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. doi: 10.18653/v1/d19-1138. URL `http://dx.doi.org/10.18653/v1/d19-1138`.

[22] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[23] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=H1eA7AEtvS`.

[24] Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HJgExaVtwr`.

[25] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL `http://arxiv.org/abs/1907.11692`.

[26] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6297–6308, 2017.

[27] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[28] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[29] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP'14*, pages 1532–1543, Doha, Qatar, October 2014. URL `http://www.aclweb.org/anthology/D14-1162`.

[30] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. *CoRR*, abs/1701.06548, 2017. URL `http://arxiv.org/abs/1701.06548`.

[31] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL*, 2018.

[32] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1493. URL `https://www.aclweb.org/anthology/P19-1493`.

[33] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. URL `https://pdfs.semanticscholar.org/cd18/800a0fe0b668a1cc19f2ec95b5003d0a5035.pdf`.

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

[35] Sebastian Ruder and Barbara Plank. Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1044–1054, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1096. URL `https://www.aclweb.org/anthology/P18-1096`.

[36] Erik Tjong Kim Sang. Introduction to the conll-2002 shared task: Language-independent named entity recognition. *CoRR*, cs.CL/0209010, 2002.

[37] Erik Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *CoNLL*, 2003.

[38] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1009. URL https://www.aclweb.org/anthology/P16-1009.

[39] Patrice Y. Simard, Yann A. LeCun, John S. Denker, and Bernard Victorri. *Transformation Invariance in Pattern Recognition — Tangent Distance and Tangent Propagation*, pages 239–274. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. ISBN 978-3-540-49430-0. doi: 10.1007/3-540-49430-8_13. URL https://doi.org/10.1007/3-540-49430-8_13.

[40] Leslie N. Smith. No more pesky learning rate guessing games. *CoRR*, abs/1506.01186, 2015. URL http://arxiv.org/abs/1506.01186.

[41] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[42] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne, 2008.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[44] Zirui Wang, Jiateng Xie, Ruochen Xu, Yiming Yang, Graham Neubig, and Jaime Carbonell. Cross-lingual alignment vs joint training: A comparative study and a simple unified framework, 2019.

[45] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101.

[46] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[47] Shijie Wu and Mark Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. 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 833–844, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1077. URL https://www.aclweb.org/anthology/D19-1077.

[48] Jiateng Xie, Zhilin Yang, Graham Neubig, Noah A. Smith, and Jaime G. Carbonell. Neural cross-lingual named entity recognition with minimal resources. *CoRR*, abs/1808.09861, 2018. URL http://arxiv.org/abs/1808.09861.

[49] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc., 2019.

[50] Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B14TlG-RW.

[51] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

[52] Zhi-Hua Zhou and Ming Li. Tri-training: exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17:1529–1541, 2005.

Appendix

A Details on distillation by clustering

One limitation of the confidence-based (single-model) distillation is that it does not consider task-specific information. Apart from classifier confidence, there could be other important features that can distinguish a good sample from a noisy one. For example, for sequence labeling, *sequence length* can be an important feature as the models tend to make more mistakes (hence noisy) for longer sequences [6]. One might also want to consider other features like *fluency*, which can be estimated by a pre-trained conditional LM like GPT [34]. In the following, we introduce a clustering-based method that can consider these additional features to separate good samples from bad ones.

Here our goal is to cluster the samples based on their *goodness*. It has been shown in computer vision that deep models tend to learn good samples faster than noisy ones, leading to a lower loss for good samples and higher loss for noisy ones [2, 15]. We propose to model *per-sample loss distribution* (along with other task-specific features) with a mixture model, which we fit using an *Expectation-Maximization* (EM) algorithm. However, contrary to those approaches which use actual (supervised) labels, we use the model predicted pseudo labels to compute the loss for the samples.

We use a two-component Gaussian Mixture Model (GMM) due to its flexibility in modeling the sharpness of a distribution [24]. In the following, we describe the EM training of the GMM for one feature, *i.e.,* per-sample loss, but it is trivial to extend it to consider other indicative task-specific features like sequence length or fluency score (see any textbook on machine learning).

EM training for two-component GMM Let $x_i \in \mathbb{R}$ denote the loss for sample $\mathbf{x}_i$ and $z_i \in \{0, 1\}$ denote its cluster id. We can write the 1d GMM model as:

$$p(x_i|\theta, \pi) = \sum_{k=0}^{1} \mathcal{N}(x_i|\mu_k, \sigma_k)\pi_k \tag{4}$$

where $\theta_k = \{\mu_k, \sigma_k^2\}$ are the parameters of the k-th mixture component and $\pi_k = p(z_i = k)$ is the probability (weight) of the k-th component with the condition $0 \leq \pi_k \leq 1$ and $\sum_k \pi_k = 1$.

In EM, we optimize the *expected complete data* log likelihood $Q(\theta, \theta^{t-1})$ defined as:

$$\begin{aligned}
Q(\theta, \theta^{t-1}) &= \mathbb{E}(\sum_i \log[p(x_i, z_i|\theta)]) && (5)\\
&= \mathbb{E}(\sum_i \sum_k \mathbb{I}(z_i = k) \log[p(x_i|\theta_k)\pi_k]) && (6)\\
&= \sum_i \sum_k \mathbb{E}(\mathbb{I}(z_i = k)) \log[p(x_i|\theta_k)\pi_k] && (7)\\
&= \sum_i \sum_k p(z_i = k|x_i, \theta^{t-1}) \log[p(x_i|\theta_k)\pi_k] && (8)\\
&= \sum_i \sum_k r_{i,k}(\theta^{t-1}) \log p(x_i|\theta_k) + r_{i,k}(\theta^{t-1}) \log \pi_k && (9)
\end{aligned}$$

where $r_{i,k}(\theta^{t-1})$ is the responsibility that cluster k takes for sample $\mathbf{x}_i$, which is computed in the E-step so that we can optimize $Q(\theta, \theta^{t-1})$ (Eq. 9) in the M-step. The E-step and M-step for a 1d GMM can be written as:

E-step: Compute $r_{i,k}(\theta^{t-1}) = \frac{\mathcal{N}(x_i|\theta_k^{t-1})\pi_k^{t-1}}{\sum_k \mathcal{N}(x_i|\theta_k^{t-1})\pi_k^{t-1}}$

M-step: Optimize $Q(\theta, \theta^{t-1})$ *w.r.t.* θ and π

- $\pi_k = \frac{\sum_i r_{i,k}}{\sum_i \sum_k r_{i,k}} = \frac{1}{N} \sum_i r_{i,k}$
- $\mu_k = \frac{\sum_i r_{i,k} x_i}{\sum_i r_{i,k}}$; $\sigma_k^2 = \frac{\sum_i r_{i,k}(x_i - \mu_k)^2}{\sum_i r_{i,k}}$

Inference For a sample $\mathbf{x}$, its *goodness* probability is the posterior probability $p(z = g|\mathbf{x}, \theta)$, where $g \in \{0, 1\}$ is the component with smaller mean loss. Here, distillation hyperparameter η is the posterior probability threshold based on which samples are selected.

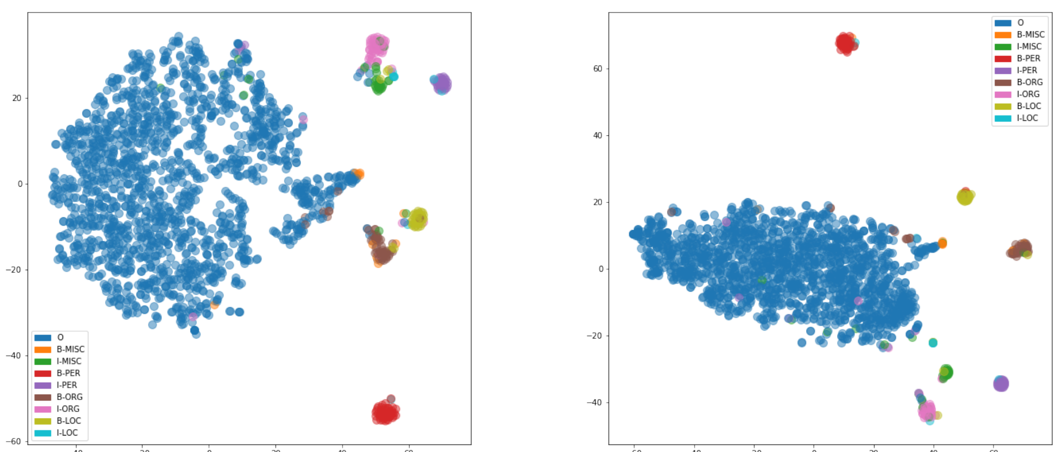
Relation with *distillation by model confidence* Astute readers might have already noticed that per-sample loss has a direct deterministic relation with the model confidence. Even though they are different, these two distillation methods consider the same source of information. However, as mentioned, the clustering-based method allows us to incorporate other indicative features like length, fluency, etc. For a fair comparison between the two methods, we use only the per-sample loss in our primary (single-model) distillation methods.

B Visualizing the effect of confidence penalty

B.1 Effect of confidence penalty in classification

In Figure 3, we present the effect of the confidence penalty (Eq. 1 in the main paper) in the target language (*Spanish*) classification on the XNER dev. data (*i.e.,* after training on English NER). We show the class distribution from the final logits (on the target language) using t-SNE plots [42].

From the figure, it is evident that the use of confidence penalty in the warm-up step makes the model more robust to unseen out-of-distribution target language data yielding better predictions, which in turn also provides a better *prior* for self-training with pseudo labels.



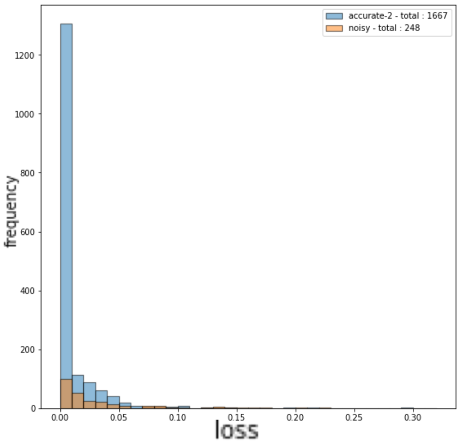
(a) Without confidence penalty.

(b) With confidence penalty.

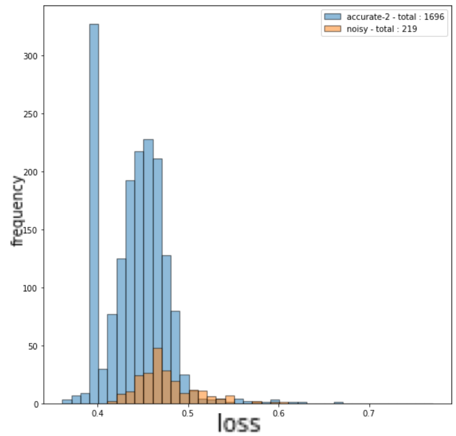
Figure 3: Effect of training with confidence penalty in the warm-up step on target (*Spanish*) language XNER classification.

B.2 Effect of confidence penalty in loss distribution

Figures 4(a) and 4(b) present the per-sample loss (*i.e.,* mean loss per sentence *w.r.t.* the pseudo labels) distribution in histogram without and with confidence penalty, respectively. Here, *accurate-2* refers to the sentences which have at most two wrong NER labels, and sentences containing more than two errors are referred to as *noisy* samples. It shows that without confidence penalty, there are many noisy samples with a small loss which is not desired. In addition to that, the figures also suggest that the confidence penalty helps to separate the clean samples from the noisy ones either by clustering or by model confidence.



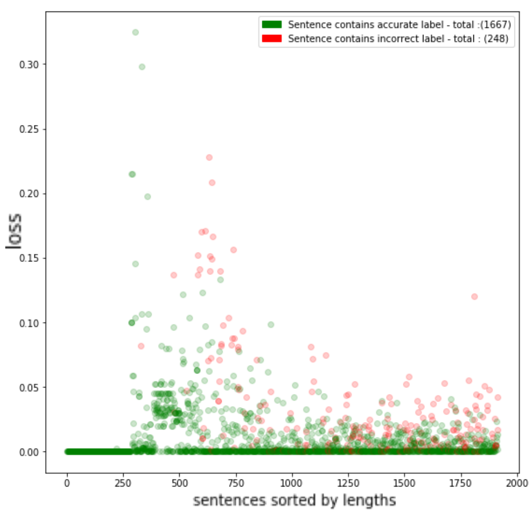
(a) Without confidence penalty.



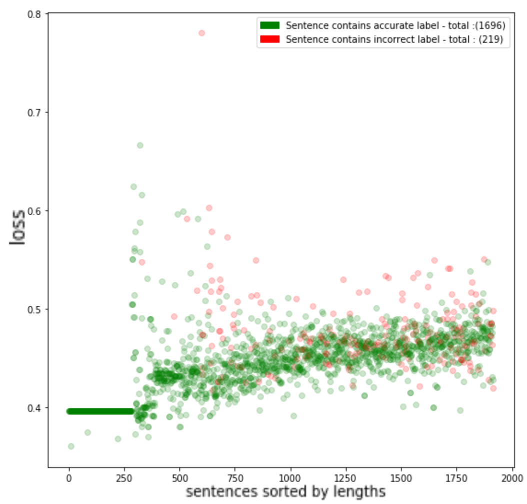
(b) With confidence penalty.

Figure 4: Histogram of loss distribution on target (*Spanish*) language XNER classification.

Figures 5(a) and 5(b) present the loss distribution in a scatter plot by sorting the sentences based on their length in the x-axis; y-axis represents the loss. As we can see, the losses are indeed more scattered when we train the model with confidence penalty, which indicates higher per-sample entropy, as expected. Also, we can see that as the sentence length increases, there are more wrong predictions. Our distillation method should be able to distill out these noisy pseudo samples.



(a) Without confidence penalty.

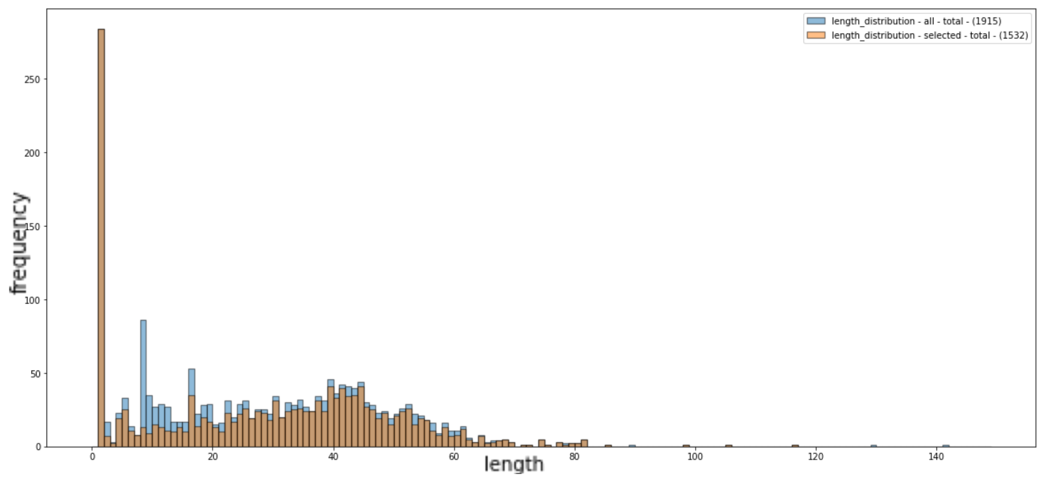


(b) With confidence penalty.

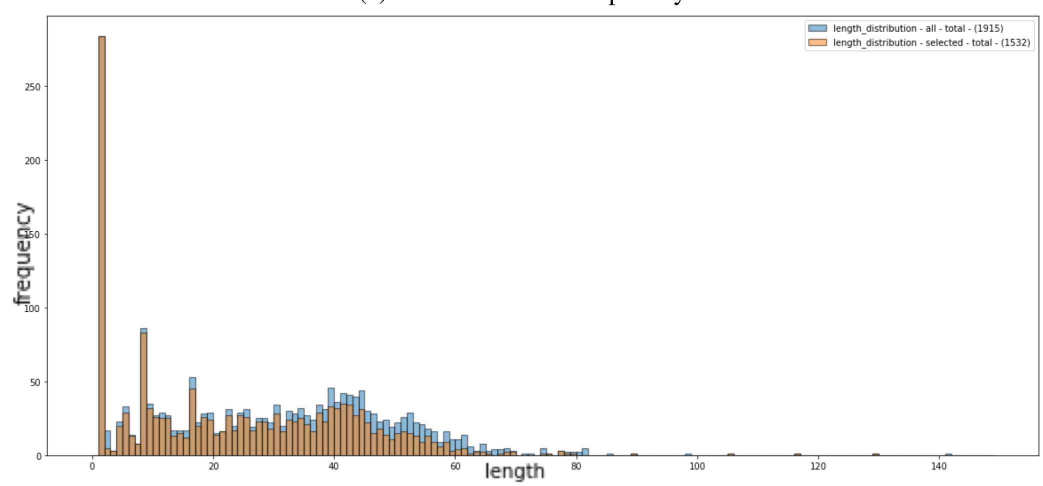
Figure 5: Scatter plot of loss distribution on target (*Spanish*) language XNER classification.

Finally, Figures 6(a) and 6(b) show the length distribution of all vs. the selected sentences (by *Distillation by model confidence*) without and with confidence penalty. From these plots, we observe that the confidence penalty also helps to perform a better distillation as more sentences are selected (by the distillation procedure) from the lower length distribution, while still covering the entire lengths.

In summary, comparing the Figures 4 - 6, we can conclude that training without confidence penalty can make the model more prone to over-fitting, resulting in more noisy pseudo labels. Training with confidence penalty not only improves pseudo labeling accuracy but also helps the distillation methods to perform better noise filtering.



(a) Without confidence penalty.



(b) With confidence penalty.

Figure 6: Distribution of selected sentence lengths on target (*Spanish*) language XNER classification.

C Extended related work

Contextual representation and cross-lingual transfer. In earlier approaches, word representations are learned from simple variants of the skip-gram model [27], where each word has a single representation regardless of its context [14, 29]. Recent approaches learn word representations that change based on the context that the word appears in [12, 17, 26, 31, 34, 49].

Peters et al. [31] propose ELMo - a bidirectional LSTM-based LM pre-training method for learning contextualized word representations. ELMo uses a linear combination of all of its layers' representations for predicting on a target task. However, because of sequential encoding, LSTM-based LM pre-training is hard to train at scale. Vaswani et al. [43] propose the Transformer architecture based on multi-headed self-attention and positional encoding. The Transformer encoder can capture long-range sequential information and allows constant time encoding of a sequence through parallelization. Radford et al. [33] propose GPT-1, which pre-trains a Transformer decoder with a conditional language model objective and then fine-tune it on the task with minimal changes to the model architecture. In the same spirit, Devlin et al. [12] propose BERT, which pre-trains a Transformer encoder with a masked language model (MLM) objective, and uses the same model architecture to adapt to a new task. The advantage of MLM objective is that it allows bidirectional encoding, whereas the standard (conditional) LM is unidirectional (*i.e.*, uses either left context or right context).

BERT also comes with a multilingual version called mBERT, which has 12 layers, 12 heads and 768 hidden dimensions, and it is trained jointly on 102 languages with a shared vocabulary of 110K subword tokens.[2] Despite any explicit cross-lingual supervision, mBERT has been shown to learn cross-lingual representations that generalise well across languages. Pires et al. [32], Wu and Dredze [47] evaluate the zero-shot cross-lingual transferability of mBERT on several NLP tasks and attribute its generalization capability to shared subword units. Pires et al. [32] additionally found structural similarity (*e.g.*, word order) to be another important factor for successful cross-lingual transfer. K et al. [20], however, show that the shared subword has minimal contribution, rather the structural similarity between languages is more crucial for effective transfer. Artetxe et al. [4] further show that joint training may not be necessary and propose an alternative method to transfer a monolingual model to a bi-lingual model through learning only the word embeddings in the target language. They also identify the vocabulary size per language as an important factor.

Lample and Conneau [22] extend mBERT with a conditional LM and a translation LM (using parallel data) objectives and a language embedding layer. They train a larger model with more monolingual data. Huang et al. [19] propose to use auxiliary tasks such as cross-lingual word recovery and paraphrase detection for pre-training. Recently, Conneau et al. [11] train the largest multilingual language model with 24-layer transformer encoder, 1024 hidden dimensions and 550M parameters. Keung et al. [21] use adversarial fine-tuning of mBERT to achieve better language invariant contextual representation for cross-lingual NER and MLDoc document classification.

Vicinal risk minimization. One of the fundamental challenges in deep learning is to train models that generalize well to examples outside the training distribution. The widely used Empirical Risk Minimization (ERM) principle where models are trained to minimize the average training error has been shown to be insufficient to achieve generalization on distributions that differ slightly from the training data [41, 51]. Data augmentation supported by the Vicinal Risk Minimization (VRM) principle [8] can be an effective choice for achieving better out-of-training generalization.

In VRM, we minimize the empirical vicinal risk defined as:

$$\mathcal{L}_v(\theta) = \frac{1}{N}\sum_{n=1}^{N} l(f_\theta(\tilde{x}_n), \tilde{y}_n) \tag{10}$$

where f_θ denotes the model parameterized by θ, and $\mathcal{D}^{\text{aug}} = \{(\tilde{x}_n, \tilde{y}_n)\}_{n=1}^{N}$ is an augmented dataset constructed by sampling the vicinal distribution $\vartheta(\tilde{x}, \tilde{y}|x_i, y_i)$ around the original training sample (x_i, y_i). Defining vicinity is however challenging as it requires to extract samples from a distribution without hurting the labels. Earlier methods apply simple rules like rotation and scaling of images [39]. Recently, Berthelot et al. [7], Zhang et al. [51] and Li et al. [24] show impressive results in image classification with simple linear interpolation of data. However, to our knowledge, none of these methods has so far been successful in NLP due to the discrete nature of texts.

[2] github.com/google-research/bert/blob/master/multilingual.md

D Setup details

D.1 Zero-shot vs. zero-resource transfer

Previous work on cross-lingual transfer has followed different training-validation standards. Xie et al. [48] perform cross-lingual transfer of NER from a source language to a target language, where they train their model on translations of the source language training data and validate it (for model selection) with target language development data. They call this as an *unsupervised setup* as they use an unsupervised word translation model [9]. Several other studies [10, 22, 44] also apply the same setting and select their model based on target language development set performance. On the other hand, Artetxe and Schwenk [3], Wu and Dredze [47] validate their models using source language development data. Bari et al. [6] show significant performance differences between validation with source vs. target language development data for NER. Later, Conneau et al. [11] provide a comprehensive analysis of different training-validation setups and encourage validating with the source language development data. Therefore, it is clear that there is no unanimous agreement regarding the proper setup. Following the previous work and landscape of the problem, we think that different settings should be considered under different circumstances.

In a pure *zero-shot* cross-lingual transfer, no target language data should be used either for training or for model selection. The goal here is to evaluate the *generalizability* and *transferability* of a model trained on a known source language distribution to an unknown target language distribution. In this sense, zero-shot setting is suitable to measure the cross-lingual transferability of a pre-trained model.

Our goal in this work is not to propose a new pre-training approach, rather to propose novel cross-lingual adaptation methods and evaluate their capability on downstream tasks. Our proposed MultiMix framework performs simultaneous self-training with data augmentation and unsupervised sample selection. As our objective is to evaluate *cross-lingual adaptation* performance and not *cross-lingual representation*, we train our model with the original source and augmented source and target language data, while validating it with target development data for model selection. We refer this as **zero-resource** setup, which is still a *minimal supervision* setting for task adaptation because no *true* target labels are used for training the model. This setup also gives us a way to compare how far we are from the supervised adaptation setting (train and validate on target language data).

D.2 Use of mBERT vs. XLM-R

From Table 4, we see that mBERT [12] trains the smallest multi-lingual language model (LM) in terms of training data size and model parameters, while XLM-R is the largest one.

Table 4: Training data size and number of model parameters of Cross-lingual Language Models.

Model Name	Tokenization	Language	#Head	#Layer	#Representation	#Vocab.	#Params.	Dataset.	Data size.
mBERT	cased	104	12	12	767	110k	172M	wiki	$\sim$ 100 GB
mBERT	uncased	102	12	12	767	110k	172M	wiki	$\sim$ 100 GB
XLM-15	uncased	15	8	12	1024	95K	250M	wiki	$\sim$ 100 GB
XLM-17	cased	17	16	16	1024	200k	570M	wiki	$\sim$ 100 GB
XLM-100	cased	100	16	16	1280	200k	570M	wiki	$\sim$ 100 GB
XLM-R	cased	100	16	16	1280	200k	570	CC-100	2.5 TB

At its heart, MultiMix uses the generation capability of a pre-trained LM for data augmentation, which could be a bottleneck for MultiMix's performance. In our initial experiments, we found that the generation quality of mBERT is not as good as that of XLM-R. Using mBERT as the vicinity model can thus generate noisy samples that can propagate to the task models and may thwart us from getting the maximum benefits from the MultiMix framework. Thus to ensure the generation of better vicinity samples, we choose to use XLM-R - the best performing multi-lingual LM to date, as the vicinity model θ_{lm} in our framework.

For the task model $\theta^{(i)}$, in principle we can use any multilingual model (*e.g.,* mBERT, XLM-R) while using XLM-R as the vicinity model. However, if we use a weaker model (*e.g.,* mBERT) compared to the vicinity model, the performance gain may not be easily distinguishable, *i.e.,* the gain may come from the increased generalization capability of the stronger vicinity model. This, in turn, can make us unable to evaluate the MultiMix framework properly in terms of its adaptation capability. In addition, from Table 1 and Table 2 (in the main paper), we observe that the zero-shot XLM-R outperforms mBERT in the *warm-up* step by $\sim 3.8\%$ in NER and $\sim 13.46\%$ in XNLI. Therefore, we choose to use XLM-R for both the task model $\theta^{(i)}$ and vicinity model θ_{lm}. Using this setup, an improvement over the baseline in MultiMix strictly indicates the superior performance of the framework.

It is also both attractive and challenging to use a single LM (XLM-R) as the vicinity model θ_{lm} over different languages. Note that the vicinity model in our framework is a disjoint pre-trained entity whose weights are not trained on any task objective. This disjoint characteristic gives our framework the flexibility to replace θ_{lm} with a better monolingual LM for a specific target language, which in turn makes our model extendable to utilize stronger and new LMs that may come in future.

E Extended dataset

XNER. For XNER, we transfer from English (en) to Spanish (es), German (de), Dutch (nl), Arabic (ar), and Finnish (fi). For English and German, we consider the dataset from CoNLL-2003 shared task [37], while for Spanish and Dutch, we use the dataset from CoNLL-2002 shared task [36]. We collected Arabic and Finnish NER datasets from Bari et al. [6]. The NER tags are converted from IOB1 to IOB2 for standardization and all the tokens of each of the six (6) datasets are classified into five (5) categories: *Person, Organization, Location, Misc.,* and *Other*. Pre-trained LMs like XLM-R generally operate at the subword level. As a result, when the labels are at the word level, if a word is broken into multiple subwords, we mask the prediction of non-first subwords. Table 5 presents the detail statsitics of the XNER datasets. We see that the datasets for different languages vary in size. Also the class-distribution is not balanced in these datasets. Therefore, we use the **micro F1 score** as the evaluation metric for XNER.

Table 5: Statistics of training, development and test datasets in different languages for XNER.

Lang	Train	Dev.	Test	XLMR data [11]	% of en
English	14041	3250	3453	300.8 GB	100
Spanish	8323	1915	1517	53.3 GB	∼17.70
Dutch	15519	2821	5076	29.3 GB	∼9.74
German	12152	2867	3005	66.6 GB	∼22.14
Arabic	2166	267	254	28.0 GB	∼9.30
Finnish	13497	986	3512	54.3 GB	∼18.05

XNLI. We use the standard XNLI dataset [10] which extends the MultiNLI dataset [45] to 15 languages. For a given pair of sentences, the task is to predict the entailment relationship between the two sentences, *i.e.*, whether the second sentence (*hypothesis*) is an *Entailment*, *Contradiction*, or *Neutral* with respect to the first one (*premise*). For XNLI, we experiment with transferring from English to Spanish (es), German (de), Arabic (ar), Swahili (sw), Hindi (hi), and Urdu (ur). Unlike NER, from Table 6, we see that the dataset sizes are same for all languages. Also the class-distribution is balanced in all the languages. Thus, we use **accuracy** as the evaluation metric for XNLI.

Table 6: Statistics of training, development and test datasets in different languages for XNLI.

Lang	Train	Dev.	Test	XLMR data [11]	% of en
English	392702	2490	5010	300.8 GB	100
Spanish	392702	2490	5010	53.3 GB	∼17.70
German	392702	2490	5010	66.6 GB	∼22.14
Arabic	392702	2490	5010	28.0 GB	∼9.30
Swahili	392702	2490	5010	1.5 GB	∼0.50
Hindi	392702	2490	5010	20.2 GB	∼6.72
Urdu	392702	2490	5010	5.7 GB	∼1.89

F Extended settings

We present the hyperparameter settings for XNER and XNLI tasks for the MultiMix framework in Table 7. In the *warm-up* step, we train and validate the task models with English data. However, for *cross-lingual adaptation*, we validate (for model selection) our model with the target language development set. We train our model with respect to the number of steps instead of the number of epochs. In the case of a given number of epochs, we convert it to a total number of steps.

Table 7: Hyperparameter settings for XNER and XNLI.

Hyperparameter	XNER		XNLI	
	Warm-up step	**Cross-lingual adaptation**	**Warm-up step**	**Cross-lingual adaptation**
		Training-hyperparameters		
model-type	`xlm-roberta-large`	`warm-up-checkpoints`	`xlm-roberta-large`	`warm-up-checkpoints`
sampling-factor α	–	0.7	–	0.7
drop-out	0.1	0.1	0.1	0.1
max-seq-length	280	280	128	128
per-gpu-train-batch-size	4	4	16	16
grad-accumulation-steps	5	4	2	2
logging-step	50	50	50	25
learning-rate (lr)	$3e^{-5}$	$5e^{-6}$	$1e^{-6}$	$1e^{-6}$
lr-warm-up-steps	200	10% of train	10% of train	10% of train
weight-decay	0.01	0.01	–	–
adam-epsilon	$1e^{-8}$	$1e^{-8}$	$1e^{-8}$	$1e^{-8}$
max-grad-norm	1.0	1.0	1.0	1.0
num-of-train-epochs	–	1	–	1
multimix-epochs	–	3	–	3
max-steps	3000	–	3000	–
train-data-percentage	100	100	5	5
conf-penalty	`True`	`False`	`True`	`False`
		Distillation-hyperparameters		
#mixture-component	–	. 2	–	–
posterior-threshold	–	0.5	–	–
covariance-type	–	`Full`	–	–
distilation-factor η	–	80, 100, 100	–	50, 80, 100
distillation-type	–	`confidence`	–	`confidence`
		Augmentation-hyperparameters		
do-lower-case	`False`	`False`	`False`	`False`
aug-type	–	`successive-max`	–	`successive-max`
aug-percentage P	–	30	–	30
diversification-factor δ	–	3	–	2×2

For both tasks, we observe that *learning rate* is a crucial hyperparameter. In table 7, *lr-warm-up-steps* refer to the *warmup-step* from triangular learning rate scheduling [40]. This hyperparameter is not to be confused with *Warm-up step* of the Multimix framework. In our experiments, *batch-size* is another crucial hyperparameter that can be obtained by multiplying per GPU training batch size with the total number of gradient accumulation steps. We fix the maximum sequence length to 280 for XNER and 128 tokens for XNLI.

For each of the experiments, we report the average score of three task models, $\theta^{(1)}$, $\theta^{(2)}$, $\theta^{(3)}$, which are initialized with different seeds. We perform each of the experiments in a single GPU setup with *float32* precision.

G Examples of augmented data

We present examples of augmented samples generated by our vicinity model for XNER and XNLI datasets in Tables 8 and 9 respectively.

Table 8: Examples of augmented data from XNER dataset.

English
Original: Motor-bike registration rose 32.7 percent in the period . **Augmented**: Motor-bike sales rose 32.7 percent in the US .
Original: He will be replaced by Eliahu Ben-Elissar , a former Israeli envoy to Egypt and right-wing Likud party politician **Augmented**: He will be led by Eliahu Cohen , a former UN Secretary to Egypt and right-wing opposition party leader .
Original: Israeli-Syrian peace talks have been deadlocked over the Golan since 1991 despite the previous government 's willingness to make Golan concessions . **Augmented**: The peace talks have been deadlocked over the Golan since 2011, despite the Saudi government 's willingness to make Golan concessions .
Spanish
Original: En esto de la comida abunda demasiado la patriotería . **Augmented**: En medio de la guerra abunda demasiado la violencia .
Original: Pero debe , cómo no , estar abierta a incorporaciones foráneas . **Augmented**: También debe , cómo no , estar abierta a personas diferentes .
Original: Deutsche Telekom calificó esta compra , cuyo precio no especificó , como otro paso hacia su internacionalización mediante adquisiciones mayoritarias destinadas a tener el control de la dirección de esas empresas . **Augmented**: Deutsche Bank calificó esta operación , cuyo importe no especificó , como otro paso hacia su expansión mediante acciones mayoritarias destinadas a tener el control de la dirección de las empresas .
Dutch
Original: Onvoldoende om een zware straf uit te spreken , luidt het . **Augmented**: Onvoldoende om een zware waarheid uit te leggen , is het .
Original: Dit hof verbindt nu geen straf aan de schuld die ze vaststelt . **Augmented**: Dit hof geeft nu de schuld aan de schuld die ze vaststelt .
Original: Wat jaren meeging als een omstreden ' CVP-dossier ' krijgt nu door de rechterlijke uitspraak het cachet van een oude koe in de gracht . **Augmented**: Wat jaren begon als een omstreden ' CVP-dossier ' krijgt nu door de rechterlijke macht het cachet van de heilige koe in de gracht .
German
Original: Gleichwohl bleibt diese wissenschaftlich abgeleitete Klassifizierung von Erzähltypen nur äußerlich . **Augmented**: Gleichwohl bleibt die daraus abgeleitete Klassifizierung von Erzähltypen nur begrenzt .
Original: Dies führt vielmehr zu anderen grundlegenden Mißverständnissen , die zur Verwischung entscheidender Unterschiede beitragen . **Augmented**: Dies führt vielmehr zu sehr großen Mißverständnissen , die zur Verwischung entscheidender Informationen führen .
Original: Die eine Geschichte zerfällt dabei in viele Erzählungen , die wiederum wissenschaftlich genau nach unterschiedlichen Genres klassifiziert werden können . **Augmented**: Die ganze Geschichte zerfällt dabei in viele Erzählungen , die nicht ganz genau in verschiedene Genres gestellt werden können .

Table 9: Examples of augmented data from XNLI dataset.

English

Original:
text_a: One of our number will carry out your instructions minutely .
text_b: A member of my team will execute your orders with immense precision .
Augmented:
text_a: One of our number will carry out your order immediately
text_b: A member of my team will execute your orders with immense care .

Original:
text_a: my walkman broke so i 'm upset now i just have to turn the stereo up real loud
text_b: I 'm upset that my walkman broke and now I have to turn the stereo up really loud .
Augmented:
text_a: my stereo broke so i 'm stuck. i just have to turn the stereo up super loud
text_b: I 'm upset because my phone broke and now I have to turn the music up really loud .

Spanish

Original:
text_a: Bueno , porque lo caliente que quiero decir como en el más frío que se pone en invierno ahí abajo , cuánto es ?
text_b: Hace calor todo el tiempo donde vivo , incluido el invierno .
Augmented:
text_a: Bueno , pero lo primero que quiero decir como en el caso calor que se pone en invierno ahí arriba , cuánto es ?
text_b: Tengo calor todo el tiempo que puedo , incluido el invierno .

Original:
text_a: Sí , es como en louisiana donde ese tipo que es como un miembro del ku klux klan algo fue elegido un poco aterrador cuando piensas en eso .
text_b: Un miembro del ku klux klan ha sido elegido en louisiana .
Augmented:
text_a: Sí , estuvieron en louisiana y ese tipo que aparece como un miembro del ku klux klan algo ha sido un poco aterrador cuando piensas en eso .
text_b: Un miembro del kumite klan ha sido detenido en louisiana .

Arabic

Original:
text_a: انه بطيء ، هناك العديد من الالات الافضل في السوق الان
text_b:. هذه اسرع الة ، لن تجدي الة افضل
Augmented:
text_a:انه صحيح ، هناك الكثير من الالات الافضل في السوق الان
text_b:. هذه اسرع طريقة ، لنجعل الة افضل

Original:
text_a:. وقد استغرق ذلك ظهور سفن النفاثة وسفن الرحلات البحرية لكي يحدث ذلك
text_b:. لا توجد سفن سياحية في المنطقة
Augmented:
text_a:. وقد سبب ذلك ظهور السيارات النفاثة وسفن الحرب البحرية لكي يحدث ذلك
text_b:. لا توجد أنشطة أخرى في المنطقة

H Acknowledgement

We would like to thank the authors and contributors of the **Transformers library** [46]. We use their implementation for XLM-R language model. We would also like to thank the authors and contributors of the **fairseq library** [28]. We use their implementation of XLM-R language model from *torch.hub*. We also thank *Alexis Conneau* for his replies in github repositories.