# Towards Fast and Accurate Modeling for Cross-Lingual Label Projection

**Thang Le[1], Huy Huu Nguyen[2], Luu Anh Tuan[3,4], Thamar Solorio[1], Thien Huu Nguyen[2]**
[1]MBZUAI, UAE [2]University of Oregon, USA
[3]Nanyang Technological University, Singapore [4]VinUniversity, Vietnam
{thang.le,thamar.solorio}@mbzuai.ac.ae
{huy,thienn}@uoregon.edu
anhtuan.luu@ntu.edu.sg

## Abstract

Information extraction (IE) systems rely on structured data for training, but such annotated data is highly imbalanced across languages, with low-resource languages receiving little attention. Label projection techniques aim to bridge this gap by transferring structured annotations from high-resource to low-resource languages. However, existing methods are either inaccurate or too slow for large-scale use. This work aims to address this problem by developing a more effective method that remains sufficiently efficient for large-scale projection. In particular, we propose to synthesize alignment sequence pairs and fine-tune an encoder model with span alignment objective, while controlling data influence during training. Experimental results across 50+ languages show that our framework consistently outperforms previous state-of-the-art methods while maintaining fast inference speed. In addition, we introduce EXP - the first benchmark for explicit evaluation of label projection, thereby reducing confounders and non-determinism in method assessment.

## 1 Introduction

The use of structured data is commonly found in downstream natural language processing (NLP) tasks such as knowledge editing (Yao et al., 2025) and relation extraction (Lai et al., 2022; Huguet Cabot et al., 2023). However, the distribution of such data is highly imbalanced across languages (Le et al., 2025), limiting the quality of trained models. As remedy, cross-lingual label projection emerged where the goal is to transfer structured annotations (e.g., named entities, relation triplets) from a high-resource source language to a lower-resource target language (Chen et al., 2023), thereby alleviating data scarcity. Despite its importance for building scalable and inclusive IE systems, existing projection methods face significant limitations: they are either too slow[1] to be practical for large-scale inference[2] or fall behind in accuracy[3] compared to the state-of-the-art (Dou and Neubig, 2021; Le et al., 2024).

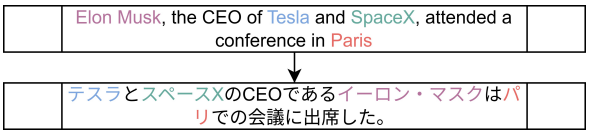


Figure 1: An example of projecting span labels from an English text to a Japanese text.

To address these challenges, we propose PVP, a new framework that bootstraps synthetic parallel data to train a specialized projection model. In particular, PVP first generates high-quality synthetic alignment pairs between source and target languages using a two-stage process: a proposer module generates candidate alignments, while a verifier module filters out potentially noisy or incorrect pairs. These refined pairs are then used to fine-tune a pre-trained language model (LM) with a span alignment objective, enabling the model to learn the mapping between structured labels across languages. Our experiments show that PVP outperforms existing state-of-the-art methods across 50+ languages, achieving a 365x speedup with respect to the previous best method while maintaining high accuracy.

Besides modeling, we also analyze the evaluation protocol adopted by previous works (Dou and Neubig, 2021; Chen et al., 2023; Le et al., 2024) i.e. *implicit evaluation*[4] and find that this protocol does not reliably reflect the quality of projected samples due to *confounding variables*, while also incurring *score variance* and *training cost*. To mitigate this, we characterize *explicit evaluation* as an

[1]See Sec. 7 for speed benchmarking
[2]Cases where the number of samples is large
[3]See Table 21 for accuracy-related benchmarking
[4]This refers to evaluating projection by training on auxiliary models



*Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 39175–39198
July 2-7, 2026 

alternative protocol for directly and efficiently evaluating label projection methods. Correspondingly, we contribute EXP - an accompanying benchmark that fully supports this protocol. EXP includes four diverse datasets across 50+ languages, providing a standardized and transparent way to assess the performance of label projection methods. Unlike prior approaches that rely on indirect evaluation through downstream tasks, EXP enables direct and stable evaluation of the projection process itself, thereby reducing confounding factors and improving reproducibility.

Our work thus advances both the methodology and evaluation standards for cross-lingual label projection, paving the way for more equitable and efficient IE systems. In summary, our main contributions are:

- We propose PVP, a new framework that bootstraps synthetic parallel samples and combine these with projection training to obtain a specialized LM for cross-lingual label projection.
- We introduce EXP, the first automatic and explicit benchmark for evaluating label projection tasks, collectively aggregating four diverse datasets in over 50 languages. The EXP benchmark thus facilitates a standardized evaluation framework that eliminates confounding variables and enables direct, stable assessment of label projection performance.
- We conduct extensive experiments to evaluate PVP against previous state-of-the-art projection methods. The results show that PVP consistently outperforms the previous state-of-the-art in both exact- and soft-matching projection scores while also having the highest throughput ratio e.g. 365x faster than the previous most accurate method.

## 2 Related Works

**Cross-Lingual Label Projection** While LLMs excel at several NLP tasks (Minaee et al., 2025), their performance on low-resource languages for IE needs further improvements (Zuo et al., 2025). To address this issue, label projection techniques have been developed aiming to project IE datasets from high-resource languages such as English to lower-resourced languages (Chen et al., 2023), enabling better coverage in model development. Initial works relied on the use of markers inserted prior to the translation phase, and extracted the corresponding text spans after translation by making use of these markers (Lewis et al., 2020; Chen et al., 2023). However, this approach suffers from decreased translation quality and occasional non-translations due to the use of markers, even upon tuning for the most performant marker pair (Chen et al., 2023; Le et al., 2024). As a remedy, (Le et al., 2024) proposed to use clean translations as templates and only search for the target span text via constrained decoding. Their approach significantly improved projection qualities while keeping translation qualities unchanged, but the runtime of their method was also significantly longer and is not scalable for projecting a large number of data samples. Concurrently, (Parekh et al., 2024) proposed to conduct projection via contextual translation, achieving promising results but still suffers from limited efficiency due to the use of LLMs.

**Word Alignment** As span alignment is closely related to word alignment, a number of projection works had also adopted word aligner in their pipelines (Stengel-Eskin et al., 2019; Eskander et al., 2020; Dou and Neubig, 2021). (Dou and Neubig, 2021) used Google Translate to translate the Spanish texts of CONLL-2002 (Tjong Kim Sang, 2002) into English, then predict the named entity tags with an English model and then project these tags back into the original Spanish texts based on word alignment mappings. Their approach was fast (by only inferring an encoder) and achieved better results than directly using the English model, but was later surpassed by more advanced projection methods such as CODEC (Le et al., 2024).

In this work, we aim to develop a projection framework that is more accurate than the previous state-of-the-art (Le et al., 2024) while maintaining fast inference speed as often seen in word alignment approaches.

## 3 Methodology

In this section, we first recall the problem setting of cross-lingual label projection. Then we introduce the pipeline for bootstrapping synthetic alignment data and the language-wise sampling strategy. Finally, we describe the projection training and inference process which makes up our PVP framework.

### 3.1 Label Projection

Formally, given an input text $x_{src}$ in the source language and $m$ span labels $\{y^i_{src} \mid i = 0, \ldots, m-1\}$, with each span $y^i_{src} = \{w^i_{src,0}, w^i_{src,1}, ..w^i_{src,l_i-1}\}$

represented by $l_i$ consecutive words in $x_{src}$.

Let the translation of $x_{src}$ in the target language be $x_{tgt}$, the goal becomes to locate span labels $\{y^i_{tgt} \mid i = 0, \ldots, m-1\}$ in $x_{tgt}$ that correspond to $\{y^i_{src} \mid i = 0, \ldots, m-1\}$ i.e. $\{y^i_{tgt} \sim y^i_{src} \mid \forall i = 0, \ldots, m-1\}$ (Figure 1).

### 3.2 Proposer-Verifier (PV)

In this section, we describe the proposer-verifier mechanism for generating alignment data. The proposer module first generates alignment proposals, which are then validated by the verifier module to be adopted in subsequent training.

**Proposer Module** To initialize the proposer module, it is straightforward to adopt a state-of-the-art projection method such as CODEC (Le et al., 2024). However, as we discussed earlier, CODEC has an extremely slow runtime and thus poses scalability issues when the number of languages and sample quantities increase. Instead, we propose to adopt a word alignment model to bootstrap proposals, which are known to be less accurate (Le et al., 2024) but are much cheaper to execute and significantly faster in practice.

Let us denote the input text as $x_{src} = [w_{src,0}, w_{src,1}..w_{src,r-1}]$ containing $r$ words, with each word $w_{src}$ represented by one or more tokens $\{t_{src,.}\}$. Correspondingly, let its translation be $x_{tgt} = [w_{tgt,0}, w_{tgt,1}..w_{tgt,u-1}]$ containing $u$ words, with each word $w_{tgt}$ represented by one or more tokens $\{t_{tgt,.}\}$. First, we aim to find a word-to-word mapping represented as the binary matrix $a^{r \times u}$ where each entry $a(i', j')$:

$$a(i', j') = \begin{cases} 1, & \text{if source word } i \text{ aligns with} \\ & \text{target word } j \\ 0, & \text{otherwise} \end{cases}$$

Traditionally in the context of label projection, AWESOME (Dou and Neubig, 2021) has often been used by previous works as a starting baseline. However, our preliminary experiments revealed that their modeling strategy is suboptimal for word alignment, which was also confirmed by recent research (Wu et al., 2023; Latouche et al., 2024). To this end, we make use of the binary alignment strategy in (Latouche et al., 2024), through which we observed better alignment qualities.

For each word $w_{src,i'}$ in $x_{src}$, we surround it with the word separator `<WORD_SEP>`, creating a new sequence $x^{i'}_{src} = [.., \texttt{<WORD\_SEP>}, w_{src,i'}, \texttt{<WORD\_SEP>}, w_{src,i'+1}, ..]$. In practice, each word is further tokenized into one or more tokens with *sentencepiece* (Kudo and Richardson, 2018) for use with pre-trained LMs. We then pass the concatenated sequence $[x^{i'}_{src}, x_{tgt}]$ through a language model $\theta$ with a sigmoid classification head to obtain probability prediction for each token $t_{tgt,j''}$ in $x_{tgt}$. Let $p^{i'}_{tgt,j''}$ be the probability prediction of the token $t_{tgt,j''}$ given $x^{i'}_{src}$. We then calculate the alignment probability between source word $i'$ and target word $j'$ as: $P(i', j') = \max(\{p^{i'}_{tgt,j''} : \forall j'' \in T(j')\})$, where $T(j')$ is the set of tokens originated from word $j'$. Here we use the $\max$ function as it was shown to perform slightly superior compared to alternatives such as the `mean` or `min` function (Latouche et al., 2024). Afterwards, we fill up the entries $a(i', j')$ in the alignment matrix based on a preset threshold $\delta$:

$$a(i', j') = \begin{cases} 1, & \text{if } P(i', j') > \delta \\ 0, & \text{otherwise} \end{cases}$$

To obtain target span projection, we adopt a matching algorithm delineated in Algorithm 1 (Appendix C). The gist is to track the consecutiveness of target word indices $\{w^{j'}_{tgt}\}$ that align with each source span $y^i_{src}$, while locating any potential conflict between projected spans $\{y^i_{tgt}\}$.

**Verifier Module** Generally, we expect that the projections obtained from the PROPOSER would contain non-trivial error rates. Thus, there is a need to construct a VERIFIER that can filter out potentially low-quality projected samples. Here we consider an approach inspired by the translation literature, which we term as BACK-PROJECTION.

We recall that the proposal procedure is *unidirectional* (or *asymmetric*). In fact, if we swap the roles of $x_{src}$ and $x_{tgt}$, we would obtain a different word alignment matrix $a_{swap}$ since the input to the language model $\theta$ would now become $[x^{j'}_{tgt}, x_{src}]$, where $j'$ is the index of each word in $x_{tgt}$. The core gist of BACK-PROJECTION is to identify highly consistent span projections $y_{tgt}$ that would remain valid even after the swap.

We delineate the proposal procedure with verification (BACK-PROJECTION) in Algorithm 2 (Appendix C). The main advantage of this approach is that there is no additional hyperparameter for the VERIFIER, making it universally adaptable to new languages and domains.

### 3.3 Language-Wise Sampling

The previous section described how to generate and verify synthetic projection labels given source texts, their span labels and corresponding parallel target texts. In this section, we denote the procedure to obtain such parallel text pairs. In particular, we choose 4 English IE datasets and make use of the `nllb-200-3.3B` (team et al., 2022) model to translate the training and development inputs to 69 typologically diverse languages. The list of dataset is shown in Table 11, and the list of languages is shown in Table 12. The datasets were chosen for their diverse domain coverage, and the languages were chosen for their broad linguistic coverage.

**Sampling Strategy** Fully translating the above datasets into the chosen languages would be rather expensive. For example, the FINDVEHICLE dataset alone contains 21.5K samples in its training split, which would require roughly 1.48M translation passes for 69 languages. Therefore, we only sample up-to $D_{train}/D_{dev}$ samples[5] (i.e. *seed inputs*) from each dataset for translation. Initially, we kept these *seed inputs* constant for all languages, but we observed that this increased repetitiveness among training data[6]. Thus, we switched to independently re-sampling the *seed inputs* for each language, thereby facilitating higher language-wise diversity during training.

### 3.4 Projection Training

We next describe our training objective upon obtaining the synthetically projected datasets. Now, each training sample $pnt$ would consist of the source sequence $x_{src}$, the target sequence $x_{tgt}$ and both source and target span labels $y^i_{src}$ and $y^i_{tgt}$.

For each source span $y^i_{src} \in x_{src}$, we surround it with the span separator `<SPAN_SEP>`, creating the new sequence $x^i_{src} = [.., \texttt{<SPAN\_SEP>}, y^i_{src}, \texttt{<SPAN\_SEP>}, ..]$. We then pass the concatenated sequence $[x^i_{src}, x_{tgt}]$ to a base language model $\phi$ and fine-tune it with the binary cross-entropy loss:

$$\mathcal{L}_{pnt} = -\frac{1}{h}\sum_{j'=g}^{g+h-1}[l_{i,j''}\log(P_\phi(i,j'')+ (1-l_{i,j''})\log(1-P_\phi(i,j''))]$$

$$l(i,j'') = \begin{cases} 1, & \text{if token } j''\text{'s original word} \in y^i_{tgt} \\ 0, & \text{otherwise} \end{cases}$$

where $g$, $h$ are the number of tokens in $x_{src}$, $x_{tgt}$.

#### 3.4.1 Controlling Point Influence

Normally, the computed losses are uniformly averaged over all data points $(x_{src}, y^i_{src}, x_{tgt}, y^i_{tgt})$, ignoring each sample's influence. In fact, we find that weighting the losses of each data point can further improve projection performance. Here we consider a simple loss weighting mechanism, termed as *worst group loss* (WGL).

**Worst Group Loss** For each batch, we average the losses over $k\%$ data points in the batch with the highest loss values (*worst group*) and only perform backpropagation on this averaged loss. Formally, we compute $\mathcal{L}_{\text{WGL},k} = \frac{1}{\lfloor Bk \rfloor}\sum_{pnt=1}^{\lfloor Bk \rfloor}\mathcal{L}_{pnt}$, where $\mathcal{L}_1 \geq \mathcal{L}_2 \geq \cdots \geq \mathcal{L}_B$ and $B$ is the number of samples present in batch. The intuition is that this prioritizes learning data points that the model is less familiar with, continually boosting its performance on the *worst group*. The detailed model update is shown in Algorithm 5 (Appendix C).

Besides WGL, we also experimented with another weighting mechanism described in Appendix B, though it was not used in the main framework as we found it to be less effective.

#### 3.4.2 Pair Augmentation

To diversify the training data, we apply two additional augmentation techniques: *bidirectional sampling* and *negative sampling*.

**Bidirectional Sampling** For label projection, the model should output the same labels for both $src2tgt$ and $tgt2src$ directions. To enforce this behaviour, we swap the roles of $src$ and $tgt$ in each data point at probability $p_b$ during training.

**Negative Sampling** Due to language mismatch and potential translation errors, correct target labels might not exist even for parallel texts. To make the model aware of such negative samples, we replace $tgt$ of each data point with a differently sampled $tgt'$ at probability $p_n$ during training. In such cases, the label entries $l(i,j'')$ are all changed to 0.

### 3.5 Inference Strategy

At inference time, we construct the hybrid alignment matrix $b^{m\times u}$, where $m$ is the number of source spans, $u$ is the number of words in $x_{tgt}$

[5] If the data split's size is less than these numbers, we take the full data split.

[6] Texts in different languages are semantically equivalent when SEED INPUTS are kept constant.

and each entry $b(i, j')$ is determined based on a preset threshold $\gamma$:

$$b(i,j') = \begin{cases} 1, & \text{if } P(i,j') > \gamma \\ 0, & \text{otherwise} \end{cases}$$

We then adopt a modified version of Algorithm 1 to obtain target span projections $y_{tgt}$. The modified process is shown in Algorithm 3 (Appendix C).

## 4 The EXP Benchmark

Before moving to the experimental section, we first discuss the evaluation strategy adopted by previous works, hereby referred to as *implicit evaluation* (Chen et al., 2023; Le et al., 2024).

**Implicit Evaluation** To evaluate the quality of a projected corpus (obtained from a projection method), previous works followed three steps. First, they fine-tuned a downstream language model $\Theta$ using the projected corpus. Second, they designate an annotated corpus $L_{test}$ in the target language and evaluate the fine-tuned model's performance on this designated corpus. Lastly, they averaged the model's performance over $k$ such fine-tuning runs[7] and treat the averaged performance as the metric for evaluating the corresponding projection method.

However, we note two major issues with the above strategy: (i) *confounding factors*: the final score now depends on external factors[8] such as the choice of $\Theta$ and $L_{test}$, and (ii) *non-determinism*: this type of evaluation reduces reproducibility as different random seeds would produce different intializations; in addition, such random seeds do not preserve across platforms, further worsening the already high variance in cross-lingual evaluation (Chen and Ritter, 2021). Moreover, the strategy incurs *expensive cost* as in order to evaluate $m$ projection methods, one would need to train and evaluate $m * k$ downstream models.

To address the above issues, we propose *explicit evaluation*, a deterministic and cost-efficient protocol for label projection assessment. Specifically, we compare the projected spans of each method against ground-truth spans using string-distance metrics[9], thereby bypassing confounding factors while preserving determinism and avoiding added training costs. To realize this protocol, we introduce the EXP benchmark, providing coverage for explicit evaluation in more than 50 languages, which we describe further below.

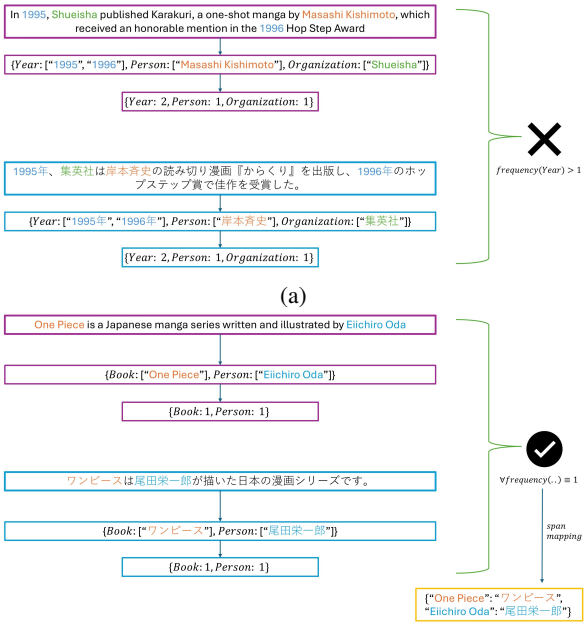


Figure 2: Examples of matching parallel samples via *label frequency dictionary* in a language pair (English-Japanese). In (2a), the matching fails as there are 2 spans of the *year* label and we do not have a reliable, autonomous way of mapping 2+ spans of the same label type. In (2b), the matching succeeds as there is exactly 1 span of each label type, and the sets of label types are equivalent in both samples. Span mappings are thus generated by matching the span in each label type of both samples.

**Explicit Evaluation** We build upon 4 corpora shown in Table 13, with each corpus covering a separate IE task to ensure coverage. These corpora were chosen as they were all constructed through translating and projecting an existing English IE corpus, where the resulting labels were verified by human experts.

For each corpus, we first aligned the parallel samples based on their metadata. Of the four, only RELX retains span-to-span mapping between the source and target texts[10] and thus obtaining groundtruth alignment labels is straightforward. For the remaining three, we conducted matching between source and target spans via using each span's label type as the anchor. Particularly, we first construct the LABEL FREQUENCY DICTIONARY (LFD) for each sample, recording the frequencies of each label type in that sample. Challenges appear when

[7] This is typically achieved by varying random seeds.
[8] We provide confounding evidence in Sec. 6
[9] We describe metric computation in Sec. 5.1
[10] Each sample in RELX features exactly one *subject* and one *object* entity labels.

there exists 2+ spans for a label type, or when LFDs of the two samples do not match. The prior would require disambiguation while the latter necessitates re-annotation, both of which cannot be reliably automated to scale to a large number of languages. Therefore, we propose to only align anchors with a single occurrence. Specifically, for two samples $x_{src}$ and $x_{tgt}$ to be aligned on the span-level, their respective LFD must not be larger than 1 for each label type, and the occurring label types must be the same for both $x_{src}$ and $x_{tgt}$[11]. We note that this still allows each sample to contain multiple span queries per text sequence, bounded by the set of label types in each dataset. Ultimately, we obtained the span alignment labels of each parallel text pair in the aligned dataset. We then aggregated the 4 datasets into a unified benchmark, which we term as EXP. Language-wise statistics are shown in Table 14-17 (Appendix A). To validate the matching heuristics, we manually examined a small subset of 50 samples in 2 language pairs (English-Japanese, English-Vietnamese) of the MASSIVE-EXP and MBABSA-EXP datasets, but did not find any erroneous case. While this only accounts for a small portion of the datasets, **we will show in the experiments that improvements observed on EXP also transfer to downstream tasks as done in previous works.**

## 5 Explicit Evaluation

### 5.1 Settings

**Hyperparameters** For $\theta$, we use an off-the-shell encoder (`Align6` (Latouche et al., 2024)) trained on six word alignment datasets including Dutch-English (Macken, 2010), Czech-English (Marek, 2008), Hindi-English (Aswani and Gaizauskas, 2005), Turkish-English (Çakmak et al., 2012), Spanish-English and Portuguese-English (Graça et al., 2008). For $\phi$, we use the `mdeberta-v3-base` model (He et al., 2021a,b). We set $\delta = \gamma = 0.5$, $D_{train} = 1000, D_{dev} = 100$. For training, we used a batch size of 64, learning rate $2e-5$ with the ADAMW optimizer (Loshchilov and Hutter, 2019), and conduct validation every 500 steps with a maximum training step of 80000. In addition, we used a linear learning-rate scheduler without warmup and set the decay length to 1000000. The checkpoint with the highest *exact matching* score on the development split is used at test time. For WGL, we set $k = 50\%$. Besides, we set $p_b = 0.5$ and $p_n = 0.2$.

**Baselines** We compare PVP with the following baselines: AWESOME (Dou and Neubig, 2021) - a word alignment method using representation similarities, WSP (Wu et al., 2023) - another word alignment method via weakly supervised pre-training, BINARY-U (Latouche et al., 2024) - the alignment strategy used in our PROPOSER, BINARY-W - a variant where the probability scores are averaged between the normal and swapped sequences, and CODEC (Le et al., 2024) - the state-of-the-art projection method based on constrained decoding. For all word alignment-based methods, we use Algorithm 1 to obtain target span projections.

**Metrics** We use *exact matching* (EM) and *soft matching* (SM) scores to evaluate each method. Given span projection $y^i_{tgt,v}$ and label $l^i_{tgt,v}$, the scores are calculated as:

$$EM = \frac{1}{n}\sum_{v \in S}(\prod_{i=1}^{q_v} y^i_{tgt,v} \equiv l^i_{tgt,v})$$

$$SM = \frac{1}{n}\sum_{v \in S}(\prod_{i=1}^{q_v} 1 - \frac{Levinshtein(y^i_{tgt,v}, l^i_{tgt,v})}{max(|y^i_{tgt,v}|, |l^i_{tgt,v}|)})$$

where $S$ is the set of projected samples, $q_v$ is the number of spans to project in sample $v$, $n$ is the total number of samples and $Levinshtein(.)$ refers to the Levinshtein distance.

### 5.2 Main Results

| Language | Awesome | WSP | Binary-U | Binary-W | Codec | PVP |
|---|---|---|---|---|---|---|
| German | 90.87 | 91.83 | 89.42 | 89.9 | 89.90 | **92.79** |
| Portuguese | 93.51 | 97.40 | 97.40 | 97.4 | 97.40 | **98.7** |
| Swedish | 94.31 | 92.31 | 94.98 | **95.32** | 89.97 | **95.32** |
| Chinese | 75.44 | 76.75 | 78.51 | 78.51 | 78.07 | **89.04** |
| Average | 88.53 | 89.57 | 90.08 | 90.28 | 88.84 | **93.96** |

Table 1: EM scores on UNER-EXP. Best score on each row is highlighted.

We first show EM scores in Table 1-4. Overall, we see that PVP significantly outperforms other methods in all 4 datasets, often by a large margin. For example, PVP performs better in RELX-EXP than the strong baseline CODEC by 4.03 EM scores on average. In MASSIVE-EXP, PVP surpasses the top baselines CODEC and BINARY-U by 6.99 and 5.87 EM scores on average. In addition, we observe that these improvements are consistent with

[11] We show examples of the matching heuristics in Figure 2

| Language | Awesome | WSP | Binary-U | Binary-W | Codec | PVP |
|---|---|---|---|---|---|---|
| Afrikaans | 82.65 | 90.48 | 90.55 | 90.34 | 90.62 | **94.4** |
| Amharic | 5.49 | 3.89 | 85.14 | 85.47 | 80.91 | **86.66** |
| Arabic | 69.43 | 73.58 | 85.47 | 85.24 | 81.96 | **88.67** |
| Azerbaijani | 55.16 | 71.43 | 85.94 | 85.79 | 70.59 | **87.55** |
| Bengali | 64.06 | 75.17 | 90.19 | 91.15 | 82.03 | **92.01** |
| Catalan | 68.25 | 71.32 | 71.86 | 71.32 | 84.59 | **90.11** |
| Welsh | 50.63 | 59.14 | 65.05 | 65.58 | **84.19** | 78.27 |
| Danish | 79.42 | 88.41 | 93.89 | 93.82 | 94.73 | **96.63** |
| German | 88.38 | 88.84 | 89.77 | 89.64 | 90.96 | **93.93** |
| Greek | 73.03 | 81.38 | 81.3 | 81.06 | 87.48 | **91.49** |
| Persian | 63.64 | 75.62 | 90.57 | 90.99 | 81.48 | **92.27** |
| Finnish | 77.55 | 84.27 | 93.99 | 93.85 | 94.41 | **96.01** |
| French | 72.84 | 75.53 | 75.66 | 75.41 | 82.98 | **87.98** |
| Hebrew | 67.79 | 75.15 | 88.96 | 88.5 | 85.28 | **90.8** |
| Hindi | 66.09 | 71.33 | 80.43 | 80.59 | 81.33 | **87.63** |
| Hungarian | 71.52 | 79.52 | 86.1 | 86.47 | 86.32 | **93.12** |
| Armenian | 69.27 | 75.69 | 90.58 | 90.58 | 77.61 | **91.72** |
| Indonesian | 56.87 | 71.34 | 83.37 | 83.15 | 83.37 | **87.53** |
| Icelandic | 61.53 | 72.14 | 83.79 | 82.68 | 87.4 | **90.71** |
| Italian | 72.88 | 75.08 | 78.06 | 77.51 | 82.92 | **87.46** |
| Javanese | 44.77 | 53.64 | 77.58 | 76.89 | **85.45** | 84.55 |
| Japanese | 61.15 | 42.29 | 73.62 | 75.2 | 47.17 | **84.52** |
| Kannada | 61.95 | 73.13 | 84.23 | 84.3 | 76.73 | **85.93** |
| Georgian | 75.00 | 75.57 | **95.04** | 94.33 | 87.68 | 94.69 |
| Khmer | 35.42 | 19.28 | 92.93 | 94.38 | 88.19 | **94.62** |
| Korean | 53.87 | 62.10 | 74.19 | 73.39 | 65.08 | **76.21** |
| Latvian | 73.35 | 78.71 | 92.57 | 92.49 | 89.93 | **97.11** |
| Malayalam | 64.07 | 70.85 | 90.59 | 91.44 | 83.98 | **92.03** |
| Mongolian | 36.11 | 53.41 | 83.02 | 82.78 | 76.11 | **83.97** |
| Malay | 53.30 | 72.82 | 88.85 | 88.2 | 87.06 | **93.33** |
| Burmese | 63.62 | 73.16 | 93.72 | **95.6** | 88.74 | 95.43 |
| Dutch | 84.89 | 87.39 | 84.56 | 84.36 | 89.88 | **92.44** |
| Norwegian | 78.50 | 88.32 | 92.28 | 91.69 | 92.02 | **95.65** |
| Polish | 73.44 | 76.36 | 81.43 | 81.43 | 82.89 | **86.72** |
| Portuguese | 67.89 | 75.88 | 74.12 | 73.17 | 86.14 | **89.66** |
| Romanian | 72.03 | 69.88 | 78.56 | 77.37 | 87.31 | **90.28** |
| Russian | 82.53 | 86.94 | 88.14 | 87.9 | 90.22 | **92.47** |
| Slovene | 75.36 | 80.46 | 88.44 | 88.2 | 89.79 | **93.14** |
| Spanish | 70.32 | 77.50 | 73.42 | 73.21 | 86.36 | **90.23** |
| Albanian | 60.36 | 68.30 | 72.23 | 71.79 | 83.12 | **85.89** |
| Swahili | 34.34 | 37.59 | 60.46 | 58.83 | **81.12** | 78.11 |
| Swedish | 85.01 | 87.73 | 92.11 | 91.68 | 91.75 | **96.34** |
| Tamil | 62.19 | 72.41 | 86.08 | 85.84 | 83.35 | **87.37** |
| Telugu | 59.60 | 72.81 | 88.7 | 89.41 | 78.1 | **90.51** |
| Tagalog | 55.54 | 61.82 | 69.09 | 68.62 | **87.38** | 83.18 |
| Thai | 45.59 | 12.79 | 79.27 | 78.06 | 65.91 | **82.35** |
| Turkish | 57.78 | 67.26 | 89.85 | 90.52 | 79.63 | **91.48** |
| Urdu | 51.11 | 62.43 | 77.87 | 78.28 | 81.26 | **84.97** |
| Vietnamese | 30.28 | 62.64 | 75.02 | 72.86 | 59.36 | **80.91** |
| Chinese | 74.51 | 73.61 | 86.6 | 85.46 | 56.37 | **89.62** |
| Average | 63.21 | 69.13 | 83.5 | 83.34 | 82.38 | **89.37** |

Table 2: EM scores on MASSIVE-EXP. Best score on each row is highlighted.

respect to individual language scores. For instance, PVP achieves highest EM scores in $4/4$ languages of RELX-EXP, $44/50$ languages of MASSIVE-EXP and $18/20$ languages of MBABSA-EXP. This shows that PVP achieves improvements on a wide range of languages and do not overfit to any specific subgroup.

To account for partial errors, we next report SM scores in Table 22-25 (Appendix F). Interestingly, we can see that CODEC is consistently the strongest baseline when measured by SM scores, with largely higher performance compared to when measured with EM scores ($5.6\%$ difference on average), suggesting that the baseline induce partial errors which gets penalized in exact matching. Nevertheless, it still gets surpassed by PVP on all

| Language | Awesome | WSP | Binary-U | Binary-W | Codec | PVP |
|---|---|---|---|---|---|---|
| Arabic | 74.62 | 83.58 | 90.46 | 90.56 | 90.51 | **94.58** |
| Danish | 95.15 | 96.58 | 98.70 | 98.61 | 98.27 | **99.22** |
| German | 91.12 | 91.83 | 93.64 | 94.04 | 94.21 | **96.64** |
| French | 74.67 | 80.45 | 84.32 | 84.71 | 89.07 | **90.49** |
| Hindi | 84.14 | 90.60 | 95.04 | 95.04 | 96.22 | **97.5** |
| Croatian | 87.16 | 91.74 | 93.04 | 93.04 | 95.98 | **97.75** |
| Indonesian | 83.99 | 89.88 | 94.44 | 94.44 | 94.48 | **97.67** |
| Japanese | 77.41 | 36.77 | 83.23 | 84.73 | 74.32 | **92.64** |
| Korean | 70.32 | 75.73 | 79.82 | 84.41 | 77.09 | **88.68** |
| Dutch | 91.79 | 93.10 | 94.45 | 94.54 | 96.16 | **97.29** |
| Portuguese | 79.36 | 83.47 | 84.36 | 84.05 | 95.0 | **95.44** |
| Russian | 87.08 | 90.72 | 92.34 | 91.99 | 95.45 | **97.15** |
| Slovak | 89.10 | 91.70 | 94.05 | 94.09 | **97.0** | 96.39 |
| Spanish | 73.28 | 77.87 | 78.98 | 79.17 | 88.31 | **89.56** |
| Swahili | 60.74 | 68.80 | 77.21 | 76.60 | **97.02** | 94.43 |
| Swedish | 94.86 | 95.12 | 98.43 | 98.43 | 97.95 | **99.0** |
| Thai | 52.59 | 28.33 | 83.49 | 83.24 | 78.11 | **87.22** |
| Turkish | 78.87 | 85.04 | 94.33 | 94.78 | 94.28 | **96.16** |
| Vietnamese | 31.37 | 80.96 | 86.45 | 85.87 | 87.57 | **90.97** |
| Chinese | 82.94 | 87.57 | 92.15 | 91.97 | 85.01 | **95.78** |
| Average | 78.03 | 80.99 | 89.45 | 89.72 | 91.1 | **94.73** |

Table 3: EM scores on MBABSA-EXP. Best score on each row is highlighted.

| Language | Awesome | WSP | Binary-U | Binary-W | Codec | PVP |
|---|---|---|---|---|---|---|
| German | 93.78 | 93.09 | 91.47 | 91.94 | 93.55 | **94.93** |
| French | 92.86 | 91.01 | 91.01 | 90.78 | 92.86 | **95.85** |
| Spanish | 84.56 | 87.56 | 85.25 | 85.02 | 87.56 | **96.08** |
| Turkish | 86.87 | 95.39 | 93.78 | 95.16 | 94.47 | **97.7** |
| Average | 89.52 | 91.76 | 90.38 | 90.72 | 92.11 | **96.14** |

Table 4: EM scores on RELX-EXP. Best score on each row is highlighted.

$4$ datasets, with notable gaps on most languages. Conclusively, **this shows that PVP achieves large improvements upon previous methods regardless of the evaluation metric.**

### 5.3 Comparison with LLMs

| Language | PVP | Codec | MIPROv2 | CLaP | SFT-small | SFT |
|---|---|---|---|---|---|---|
| Arabic | **97.13** | 94.26 | 72.76 | 83.40 | 93.53 | 96.02 |
| Danish | **99.71** | 98.83 | 81.26 | 86.96 | 96.74 | 99.27 |
| German | **98.6** | 97.14 | 77.92 | 88.65 | 96.84 | 97.77 |
| French | **96.02** | 95.06 | 71.07 | 83.45 | 93.24 | 94.33 |
| Hindi | **98.94** | 98.32 | 72.43 | 92.25 | 90.26 | 97.77 |
| Croatian | **98.96** | 97.78 | 76.95 | 83.28 | 91.60 | 97.53 |
| Indonesian | **98.81** | 96.62 | 73.51 | 87.58 | 96.18 | 98.26 |
| Japanese | **96.25** | 89.93 | 61.08 | 89.91 | 91.12 | 93.98 |
| Korean | **93.96** | 85.57 | 61.03 | 82.67 | 88.53 | 93.05 |
| Dutch | **98.7** | 97.89 | 81.17 | 86.95 | 97.26 | 97.80 |
| Portuguese | **97.89** | 97.6 | 73.97 | 88.28 | 93.92 | 96.18 |
| Russian | **98.66** | 97.51 | 77.89 | 85.98 | 96.21 | 97.87 |
| Slovak | **98.38** | 98.34 | 76.55 | 86.26 | 93.38 | 97.55 |
| Spanish | **95.46** | 94.32 | 70.72 | 86.34 | 91.99 | 93.95 |
| Swahili | 97.02 | **98.22** | 70.08 | 82.32 | 65.13 | 81.95 |
| Swedish | **99.62** | 98.76 | 81.38 | 88.99 | 96.95 | 99.23 |
| Thai | **92.94** | 89.26 | 63.60 | 85.27 | 89.39 | 92.17 |
| Turkish | **97.93** | 96.56 | 68.14 | 82.87 | 89.86 | 97.49 |
| Vietnamese | **95.31** | 93.39 | 33.38 | 86.06 | 84.80 | 93.26 |
| Chinese | **98.09** | 93.23 | 73.55 | 89.85 | 95.92 | 97.32 |
| Average | **97.42** | 95.43 | 70.92 | 86.37 | 91.64 | 95.64 |

Table 5: SM scores (including LLMs) on MBABSA-EXP. Best score on each row is highlighted.

As there are increased interests in the use of LLMs recently (Matarazzo and Torlone, 2025), we

| Language | PVP | Codec | MIPROv2 | CLaP | SFT-small | SFT |
|---|---|---|---|---|---|---|
| German | **98.22** | 97.25 | 58.35 | 44.53 | 96.32 | 97.53 |
| French | **98.03** | 96.22 | 47.16 | 29.52 | 95.90 | 96.88 |
| Spanish | **98.7** | 94.42 | 44.44 | 54.18 | 94.24 | 96.80 |
| Turkish | **98.79** | 95.54 | 51.94 | 46.40 | 90.17 | 96.40 |
| Average | **98.44** | 95.86 | 50.47 | 43.66 | 94.16 | 96.90 |

Table 6: SM scores (including LLMs) on RELX-EXP. Best score on each row is highlighted.

| Language | PVP | Codec | MIPROv2 (70B) | CLaP (70B) | SFT-Large (14B) |
|---|---|---|---|---|---|
| German | **98.22** | 97.25 | 83.19 | 70.24 | 97.46 |
| French | **98.03** | 96.22 | 85.66 | 61.30 | 97.68 |
| Spanish | **98.7** | 94.42 | 79.93 | 72.46 | 98.48 |
| Turkish | **98.79** | 95.54 | 70.24 | 70.89 | 97.94 |
| Average | **98.44** | 95.86 | 79.75 | 68.72 | 97.89 |

Table 7: SM scores (scaling LLMs) on RELX-EXP. Best score on each row is highlighted.

are interested in how our (encoder-only) framework would fare against them. To investigate, we setup three baselines involving LLMs: MIPROv2 - an in-context learning baseline that optimizes a few-shot prompt using the framework proposed in (Opsahl-Ong et al., 2024), CLAP - another in-context baseline that projects labels via contextual translation with LLMs (Parekh et al., 2024), SFT-SMALL - a supervised fine-tuning baseline that trains a small LLM on the synthetic data bootstrapped in PVP, and SFT - a similar baseline but makes use of a larger LLM. As backbones, we use the QWEN2.5-0.5B-INSTRUCT and QWEN2.5-3B-INSTRUCT (Yang et al., 2025) models for SFT-SMALL and SFT. Regarding MIPROv2[12], we initially experimented with the QWEN2.5-7B-INSTRUCT model but found that the LLAMA-3.1-8B-INSTRUCT (Grattafiori et al., 2024) model achieved better results and thus opted for the latter. We report results on MBABSA-EXP and RELX-EXP in Table 5 and 6[13]. We find that compared to previous state-of-the-art (CODEC), MIPROv2 and CLAP perform largely inferior while SFT-SMALL and SFT perform quite competitively. Nevertheless, **PVP still performs better than baselines on most languages**, signaling that **our method is still strong even when compared with the evaluated LLM methods**.

**Scaling parameters** Interestingly, we observed that SFT consistently outperformed SFT-SMALL, suggesting that scaling LLM's size also improves projection performance. Thus, we conducted another comparison with the LLM methods, but this time using larger size LLMs. Particularly, we use LLAMA-3.3-70B-INSTRUCT for in-context baselines (MIPROv2/CLAP) and QWEN2.5-14B-INSTRUCT for fine-tuning (SFT-LARGE). Due to expensive inference cost, we only evaluated these baselines on the RELX-EXP dataset. Results are shown in Table 7. As expected, both SFT and few-shot methods benefit from scaling to larger sizes (8B/3B → 70B/14B). However, even with these substantially larger models, performance still remains below PVP. Notably, **PVP only contains 278M parameters, significantly smaller than the fine-tuned 14B LLM (SFT-LARGE) that still performs worse.**

## 6 Implicit Evaluation

| Language | Codec | PVP | English-FT |
|---|---|---|---|
| Afrikaans | 69.3 | **70.8** | 64.1 |
| Amharic | 57.3 | **58.7** | 29.9 |
| Egyptian Arabic | 60.2 | **64.6** | 53.3 |
| South Azerbaijani | 38.2 | 40.7 | **49.3** |
| Bengali | **67.5** | 66.9 | 51.7 |
| Catalan | 65.1 | **68.1** | 57.4 |
| Welsh | 53.2 | **55.2** | 32.9 |
| Danish | 70.9 | **73.5** | 66.7 |
| German | 69.0 | **71.4** | 70.5 |
| Greek | 66.2 | **70.0** | 64.4 |
| Western Persian | 67.4 | **70.3** | 61.3 |
| Finnish | 64.1 | **67.6** | 58.0 |
| French | 65.7 | **70.6** | 61.5 |
| Hebrew | 61.9 | **65.7** | 43.0 |
| Hindi | **66.1** | 65.7 | 55.8 |
| Hungarian | 64.4 | **67.3** | 60.8 |
| Armenian | 60.1 | **64.1** | 52.4 |
| Indonesian | 64.5 | **68.3** | 60.8 |
| Icelandic | 60.7 | **65.2** | 51.7 |
| Italian | 69.4 | **70.4** | 63.5 |
| Average | 63.06 | **65.75** | 55.45 |

Table 8: F1 scores of 20 languages in MASSIVE (downstream performance with MDEBERTA-V3-BASE). Best score on each row is highlighted.

| Language | Backbone | Codec | PVP |
|---|---|---|---|
| South Azerbaijani | XLM-R | 40.00 | 38.00 |
| Bengali | XLM-R | 66.10 | 66.40 |
| South Azerbaijani | MDEBERTA | 38.20 | 40.70 |
| Bengali | MDEBERTA | 67.50 | 66.90 |

Table 9: Conflicts arise in F1 scores when varying the backbone model on two languages in MASSIVE.

To measure improvements on downstream task, we further fine-tune LMs using the target data projected by each method. Here we use the MDEBERTA-V3-BASE model and average test results over 5 distinct random seeds. We compare PVP with two baselines: CODEC and ENGLISH-FT (fine-tune on English training data). Models

[12]We set the number of demonstrations as $k = 3$.

[13]For conciseness's sake, we only show SM scores here (analyses are similar with EM scores). Results on the remaining two datasets are placed in Appendix F.3.

are evaluated on the Slot Filling task using 20 languages in the MASSIVE dataset. Results are shown in Table 8. As can be seen, models trained on PVP's data perform better than baselines on $17/20$ languages, and improve by $10.25\%$ on average compared to the zero-shot English models. In addition, this empirically proves that improvements in explicit evaluation transfers to the implicit setting, showing that **our evaluation results align with previous works' settings while being cheaper**[14] **to conduct and directly relevant to label projection** itself.

**Changing backbone** To isolate the effect of backbone choice, we additionally experimented with using the XLM-ROBERTA-BASE model for downstream fine-tuning. We maintain the same hyperparameters and seeds as used with MDEBERTA-V3-BASE. Results are shown in Table 28 (Appendix F.4). As can be seen, PVP still facilitate better downstream performance than baselines on most languages. However, these results also reveal **confounding evidence on two languages that arise due to the backbone switch in implicit evaluation**, which we discuss further below.

Ideally, given two projected corpora A and B, the binary conclusion — whether A is of higher quality than B — should not depend on external factors such as the downstream model. In practice, as highlighted in Table 9, this conclusion can change when we vary the downstream model. When using XLM-ROBERTA-BASE as the backbone, CODEC performs better on South Azerbaijani and slightly worse on Bengali. However, when switching to MDEBERTA-V3-BASE, the relative outcome is reversed. This means the ranking between projection methods now depends on the downstream model choice, even though the projected data itself is fixed. In contrast, **this issue does not arise in explicit evaluation**, where we directly compare projected labels with ground-truth annotations and do not involve downstream training.

## 7 Efficiency Evaluation

To inspect improvements in inference speed, we use CONLL-2003 (Tjong Kim Sang and De Meulder, 2003) as the source dataset (which contains 10K English samples with labeled spans) and measure time taken for each method to project these samples into 5 low-resource languages. We show these measures in Table 10, and provide setting details in Appendix H. We see that AWESOME achieves fastest speed, closely followed by PVP. Theoretically, in terms of forward passes, both BINARY and WSP scale in proportionate to $O(n * k)$ but with distinct batching mechanisms, where $k$ is the number of words in the sequence and $n$ is the number of samples. Meanwhile, AWESOME only conducts one forward pass per sample, thus making it the fastest method ($O(n)$). CODEC relies on constrained decoding with autoregressive models and is thus the slowest method. In contrast, PVP scales in $O(n * m)$, where $m$ is the number of labeled spans to project, making it the second fastest [15]. However, we note that PVP is the most accurate method (Table 21, Appendix F.1) and thus achieves good speed-performance balance, compared to AWESOME which achieves lowest EM/SM scores despite being fastest. **Compared to the second most performant method CODEC, PVP is at least $365\times$ faster while also being more accurate**[16].

| Language | Awesome | WSP | Binary-U | Binary-W | Codec | PVP |
|---|---|---|---|---|---|---|
| Bambara | **19.07** | 493.94 | 739.47 | 1216.45 | 17522.84 | 43.98 |
| Ewe | **19.07** | 494.58 | 774.36 | 1286.77 | 19065.42 | 44.32 |
| Fon | **20.16** | 578.22 | 946.33 | 1579.52 | 26461.28 | 53.25 |
| Hausa | **18.64** | 472.04 | 684.72 | 1127.41 | 15092.94 | 38.71 |
| Igbo | **19.74** | 502.41 | 742.75 | 1212.30 | 15511.31 | 42.45 |

Table 10: Time taken (in seconds) for each method to project 10K English samples from CONLL-2003 to 5 languages. We highlight the fastest method and underline the second.

## 8 Conclusion

In this paper, we aim to construct a new framework for cross-lingual label projection that achieves better balance in performance and efficiency. Specifically, our framework featured construction of synthetic alignment data coupled with projection training to build a specialized encoder for label projection. In addition, we constructed EXP - the first benchmark for automatic explicit evaluation of label projection methods, cumulatively covering 50+ languages. Lastly, we conduct extensive experiments to benchmark our framework against previous state-of-the-art projection methods (including LLMs), where we observed consistent improvements both in exact- and soft-matching scores while remaining extremely fast.

[14] Explicit evaluation does not require downstream task fine-tuning.

[15] In most samples, we have $m \ll k$

[16] The $365\times$ speedup is relative to CODEC on the Igbo language, where improvement is the most modest.

## Limitations

In this work, we constructed the EXP benchmark by preprocessing and aligning multilingual samples from 4 existing datasets. Of these, only RELX is *gold-standard* as each parallel pair retains exact span-to-span mappings by human annotators. For the other three, we had to adopt a matching heuristics and thus their qualities remain at *silver-standard*. While we made efforts to inspect a small subset of matched samples, we note that $3/4$ datasets in EXP were not fully validated. Still, we believe that the benchmark would help reduce experiment cost and confounders in evaluating label projection methods. As shown in our experiments, methods performing well on EXP also perform well when evaluated in the same setting (implicit) as done by previous works.

Additionally, although we conducted experiments over a wide range of languages and datasets, these only cover a small subset of the 500+ institutional languages available (Bird, 2024). Therefore, the results in this work should be interpreted as within the described experiment scope, and we do not claim that PVP will definitely deliver strong results on all languages and datasets.

## Acknowledgements

We thank the anonymous reviewers for their helpful and constructive feedback.

## References

Niraj Aswani and Robert Gaizauskas. 2005. Aligning words in English-Hindi parallel corpora. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 115–118, Ann Arbor, Michigan. Association for Computational Linguistics.

Steven Bird. 2024. Must NLP be extractive? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14915–14929, Bangkok, Thailand. Association for Computational Linguistics.

Mehmet Talha Çakmak, Süleyman Acar, and Gülşen Eryiğit. 2012. Word alignment for English-Turkish language pair. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2177–2180, Istanbul, Turkey. European Language Resources Association (ELRA).

Pei Chen, Haotian Xu, Cheng Zhang, and Ruihong Huang. 2022. Crossroads, buildings and neighborhoods: A dataset for fine-grained location recognition. In *NAACL*. Association for Computational Linguistics.

Yang Chen, Chao Jiang, Alan Ritter, and Wei Xu. 2023. Frustratingly easy label projection for cross-lingual transfer. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5775–5796, Toronto, Canada. Association for Computational Linguistics.

Yang Chen and Alan Ritter. 2021. Model selection for cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5675–5687, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Ramy Eskander, Smaranda Muresan, and Michael Collins. 2020. Unsupervised cross-lingual part-of-speech tagging for truly low-resource scenarios. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4820–4831, Online. Association for Computational Linguistics.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2023. MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4277–4302, Toronto, Canada. Association for Computational Linguistics.

João Graça, Joana Paulo Pardal, Luísa Coheur, and Diamantino Caseiro. 2008. Building a golden collection of parallel multi-language word alignment. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong,

Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr

Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Runwei Guan, Ka Lok Man, Feifan Chen, Shanliang Yao, Rongsheng Hu, Xiaohui Zhu, Jeremy Smith, Eng Gee Lim, and Yutao Yue. 2024. Findvehicle and vehiclefinder: a ner dataset for natural language-based vehicle retrieval and a keyword-based cross-modal vehicle retrieval system. *Multimedia Tools and Applications*, 83(8):24841–24874.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

Pere-Lluis Huguet Cabot, Simone Tedeschi, Axel-Cyrille Ngonga Ngomo, and Roberto Navigli. 2023. Redfm: a filtered and multilingual relation extraction dataset. In *Proc. of the 61st Annual Meeting of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada. Association for Computational Linguistics.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. Dspy: Compiling declarative language model calls into self-improving pipelines.

Abdullatif Koksal and Arzucan Ozgur. 2020. The RELX dataset and matching the multilingual blanks for cross-lingual relation classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 340–350, Online. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Viet Lai, Hieu Man, Linh Ngo, Franck Dernoncourt, and Thien Nguyen. 2022. Multilingual SubEvent relation extraction: A novel dataset and structure induction method. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5559–5570, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Gaetan Latouche, Marc-André Carbonneau, and Benjamin Swanson. 2024. BinaryAlign: Word alignment as binary sequence labeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10277–10288, Bangkok, Thailand. Association for Computational Linguistics.

Duong Minh Le, Yang Chen, Alan Ritter, and Wei Xu. 2024. Constrained decoding for cross-lingual label projection. In *The Twelfth International Conference on Learning Representations*.

Thang Le, Huy Huu Nguyen, Anh Tuan Luu, and Thien Huu Nguyen. 2025. Massively multilingual instruction-following information extraction. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3542–3585, Vienna, Austria. Association for Computational Linguistics.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In

*Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2020. Crossner: Evaluating cross-domain named entity recognition. In *AAAI Conference on Artificial Intelligence*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Lieve Macken. 2010. An annotation scheme and gold standard for Dutch-English word alignment. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

David Marek. 2008. Automatic alignment of tectogrammatical trees from czech-english parallel corpus.

Andrea Matarazzo and Riccardo Torlone. 2025. A survey on large language models with some insights on their capabilities and limitations. *Preprint*, arXiv:2501.04040.

Stephen Mayhew, Terra Blevins, Shuheng Liu, Marek Suppa, Hila Gonen, Joseph Marvin Imperial, Börje Karlsson, Peiqin Lin, Nikola Ljubešić, Lester James Miranda, Barbara Plank, Arij Riabi, and Yuval Pinter. 2024. Universal NER: A gold-standard multilingual named entity recognition benchmark. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4322–4337, Mexico City, Mexico. Association for Computational Linguistics.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2025. Large language models: A survey. *Preprint*, arXiv:2402.06196.

Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. 2024. Optimizing instructions and demonstrations for multi-stage language model programs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9340–9366, Miami, Florida, USA. Association for Computational Linguistics.

Tanmay Parekh, I-Hung Hsu, Kuan-Hao Huang, Kai-Wei Chang, and Nanyun Peng. 2024. Contextual label projection for cross-lingual structured prediction. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5738–5757, Mexico City, Mexico. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Elias Stengel-Eskin, Tzu-ray Su, Matt Post, and Benjamin Van Durme. 2019. A discriminative neural model for cross-lingual word alignment. 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 910–920, Hong Kong, China. Association for Computational Linguistics.

Nllb team, Marta Ruiz Costa-jussà, James Cross, Onur cCelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Alison Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon L. Spruit, C. Tran, Pierre Yves Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzm'an, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *ArXiv*, abs/2207.04672.

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)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chengyan Wu, Bolei Ma, Yihong Liu, Zheyu Zhang, Ningyuan Deng, Yanshu Li, Baolan Chen, Yi Zhang,

Barbara Plank, and Yun Xue. 2025. M-absa: A multilingual dataset for aspect-based sentiment analysis. *Preprint*, arXiv:2502.11824.

Qiyu Wu, Masaaki Nagata, and Yoshimasa Tsuruoka. 2023. WSPAlign: Word alignment pre-training via large-scale weakly supervised span prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11084–11099, Toronto, Canada. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Yunzhi Yao, Jizhan Fang, Jia-Chen Gu, Ningyu Zhang, Shumin Deng, Huajun Chen, and Nanyun Peng. 2025. CaKE: Circuit-aware editing enables generalizable knowledge learners. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 11377–11393, Suzhou, China. Association for Computational Linguistics.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. 2024. Swift:a scalable lightweight infrastructure for fine-tuning. *Preprint*, arXiv:2408.05517.

Yuxin Zuo, Wenxuan Jiang, Wenxuan Liu, Zixuan Li, Long Bai, Hanbin Wang, Yutao Zeng, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2025. Knowcoder-x: Boosting multilingual information extraction via code. *Preprint*, arXiv:2411.04794.

# A Data Statistics

List of languages and datasets are shown in Table 11, 12, 13. Data statistics are shown in Table 14, 15, 16, 17.

| Dataset | Domain |
|---|---|
| CrossNER (Liu et al., 2020) | Politics, Natural Science, Music, Literature, Artificial Intelligence |
| FindVehicle (Guan et al., 2024) | Traffic |
| HarveyNER (Chen et al., 2022) | Social Media, Disaster, Geolocation |
| Ontonotes5 (Hovy et al., 2006) | News, Blog, Dialogue |

Table 11: List of IE datasets for bootstrapping synthetic projection labels

# B Learnable Group Loss

We describe an alternative loss weighting mechanism for modulating sample influence during training, termed as *learnable group loss* (LGL).

**Learnable Group Loss** We initialize learnable language weight $q^l = \frac{1}{|L|}$ for each language $l \in L$, where $L$ is the language set excluding English[17]. These weights are then used to adjust the influence of each language's loss and trained end-to-end together with $\phi$. The detailed model update is described in Algorithm 4 (Appendix C). In our experiments, we observe that both variants can improve performance compared to uniform averaging, with WGL achieving better results. Thus, we selected WGL as the weighting mechanism for our framework.

# C Algorithm List

We outline specific implementations in Algorithm 1, 2, 3, 4, 5 (referred to in the main paper).

# D Statistics on Alignment Heuristics

In Table 18-20, we report filtering statistics in terms of retained samples, retained spans, and average text length before and after filtering. Overall, MASSIVE-EXP retains more than $90\%$ of labeled samples and spans. MBABSA-EXP applies a stricter filter, retaining $74-85\%$ of samples and around $60\%$ of spans. UNER-EXP is the most restrictive, retaining $22-30\%$ of samples and roughly $30\%$ of spans. Importantly, the average text length remains very similar before and after filtering across all three datasets, with a maximum difference of about 4 words (observed in MBABSA-EXP). This suggests that the filtering does not sys-

[17] In this work, we focus on projecting data between English and other languages as English is the highest resourced language. Thus, the number of groups is reduced to $|L|$ instead of $\frac{|L|*(|L|+1)}{2}$.

**Algorithm 1:** Span projection from word alignment

**Input:**

$a^{r \times u}$ - word alignment matrix;

$x_{src} = [w^0_{src}, w^1_{src}..w^{r-1}_{src}]$ - source sequence;

$x_{tgt} = [w^0_{tgt}, w^1_{tgt}..w^{u-1}_{tgt}]$ - target sequence;

$y_{src} = \{y^i_{src} \mid i = 0, \ldots, m-1\}$ - list of source spans;

**Output:**

$y_{tgt} = \{y^i_{tgt} \mid i = 0, \ldots, m-1\}$ - list of projected target spans;

---

$shared_{src} \leftarrow \{\}$ ; /* Initialize set to track conflicting word indices in $y_{tgt}$ */

**for** $y^i_{src} \in y_{src}$ **do**

  $s^i_{src} \leftarrow \{\}$ ; /* Initialize set to track word indices in $x_{tgt}$ that align with at least one word in $y^i_{src}$ */

  **for** $w^{i'}_{src} \in y^i_{src}$ **do**

    **for** $j' = 0$ **to** $u - 1$ **do**

      **if** $a(i', j') \equiv 1$ **then**

        $s^i_{src} \leftarrow s^i_{src} \cup j'$;

      **end**

    **end**

  **end**

  $convert\_to\_list(s^i_{src})$;

  $sort\_in\_ascending\_order(s^i_{src})$;

  **if** $is\_consecutive(s^i_{src})$ ***and*** $non\_empty(s^i_{src})$ ***and*** $s^i_{src} \cap shared_{src} \equiv \{\}$ ; /* Span indices need to be consecutive and non-conflicting */ **then**

    $y^i_{tgt} \leftarrow s^i_{src}$;

    $shared_{src} \leftarrow shared_{src} \cup s^i_{src}$;

  **else**

    $exit()$ ; /* Projection failed, algorithm stops */

  **end**

**end**

**Algorithm 2:** Span projection with back-projection

**Input:**
$x_{src} = [w^0_{src}, w^1_{src}..w^{r-1}_{src}]$ - source sequence;
$x_{tgt} = [w^0_{tgt}, w^1_{tgt}..w^{u-1}_{tgt}]$ - target sequence;
$y_{src} = \{y^i_{src} \mid i = 0, \ldots, m-1\}$ - list of source spans;
**Output:**
$y_{tgt} = \{y^i_{tgt} \mid i = 0, \ldots, m-1\}$ - list of projected target spans;

---

$a \leftarrow word\_alignment(x_{src}, x_{tgt})$;
$a_{swap} \leftarrow word\_alignment(x_{tgt}, x_{src})$ ; /* Swap roles */
$s_{tgt} \leftarrow span\_projection\_from\_word\_alignment(a, x_{src}, x_{tgt})$;
$s_{tgt,swap} \leftarrow span\_projection\_from\_word\_alignment(a_{swap}, x_{src}, x_{tgt})$;
; /* Projection succeeded for both normal and swapped roles, move on to verify */
**for** $i = 0$ **to** $m - 1$ **do**
  **if** $s^i_{tgt} \equiv s^i_{tgt,swap}$ **then**
    $y^i_{tgt} \leftarrow s^i_{tgt}$;
  **else**
    $exit()$ ; /* Verification failed, algorithm stops */
  **end**
**end**

tematically bias toward shorter or simpler texts. In short, the heuristic alignment affects each dataset differently, rather than uniformly shifting them toward any artificially "easier" shared distribution.

## E Contamination Check

To inspect potential overlap with the synthetic training datasets (Table 11), we conducted measurements in two ways. First, we computed sample-level text overlap between the English portions of the synthetic training datasets and EXP evaluation datasets (to account for possible cross-lingual leakage) and found $0\%$ overlap. Second, we measured overlap at the labeled span level (e.g., named entities), which was $14.2\%$. This is relatively low, especially compared to a standard gold IE benchmark such as CoNLL2003 (Tjong Kim Sang and De Meulder, 2003), where the span overlap ratio is $52\%$. We also note that the synthesized training data and the evaluation datasets were drawn from different sources (Table 11, 13), which explains their minimal overlap.

## F Additional Results

### F.1 Explicit Results on average

We show average of $EM/SM$ scores over all datasets in EXP in Table 21. Here PVP performs better than baselines on both metrics.

### F.2 Explicit Results in SM scores

We show SM scores in Table 22, 23, 24, 25. The improvements are consistent with what we observed in EM scores.

### F.3 Explicit Results with LLMs in SM scores

We show LLMs results on MASSIVE-EXP and UNER-EXP in Table 26 and 27. We observe that SFT achieves strong results in UNER-EXP, performing better than other methods in $2/4$ languages. Still, both SFT and SFT-SMALL underperforms in MASSIVE-EXP compared to PVP with a $2.84\%$ difference in SM scores on average. While we find LLM-based methods to perform competitively in some cases (e.g. UNER-EXP), they usually underperform our framework in most languages/datasets despite being more expensive to execute.

### F.4 Implicit Evaluation with backbone change

We show results with XLM-ROBERTA-BASE in Table 28. Here PVP's projected data still facilitates better downstream performance than baselines on most languages.

**Algorithm 3:** Span projection from hybrid alignment

**Input:**
$b^{m \times u}$ - hybrid alignment matrix;
$x_{src} = [w^0_{src}, w^1_{src}..w^{r-1}_{src}]$ - source sequence;
$x_{tgt} = [w^0_{tgt}, w^1_{tgt}..w^{u-1}_{tgt}]$ - target sequence;
$y_{src} = \{y^i_{src} \mid i = 0, \ldots, m-1\}$ - list of source spans;
**Output:**
$y_{tgt} = \{y^i_{tgt} \mid i = 0, \ldots, m-1\}$ - list of projected target spans;

---

$shared_{src} \leftarrow \{\}$ ; /* Initialize set to track conflicting word indices in $y_{tgt}$ */
**for** $i = 0$ **to** $m - 1$ **do**
  $s^i_{src} \leftarrow \{\}$ ; /* Initialize set to track word indices in $x_{tgt}$ that align with $y^i_{src}$ */
  **for** $j' = 0$ **to** $u - 1$ **do**
    **if** $b(i, j') \equiv 1$ **then**
      $s^i_{src} \leftarrow s^i_{src} \cup j'$;
    **end**
  **end**
  $convert\_to\_list(s^i_{src})$;
  $sort\_in\_ascending\_order(s^i_{src})$;
  **if** $is\_consecutive(s^i_{src})$ ***and*** $non\_empty(s^i_{src})$ ***and*** $s^i_{src} \cap shared_{src} \equiv \{\}$ ; /* Span indices need to be consecutive and non-conflicting */ **then**
    $y^i_{tgt} \leftarrow s^i_{src}$;
    $shared_{src} \leftarrow shared_{src} \cup s^i_{src}$;
  **else**
    $exit()$ ; /* Projection failed, algorithm stops */
  **end**
**end**

**Algorithm 4:** Fine-tuning with Learnable Group Loss

**Input:**
$L$ - language set;
$\theta$ - base model;
$opt$ - optimizer;
$sched$ - learning rate scheduler;
$d_{train}$ - training data;
**Output:**
$\theta_{ft}$ - fine-tuned model with learnable group loss;

---

**for** $l \in L$ **do**
  $q_l \leftarrow \frac{1}{|L|}$;
**end**
$\theta_{ft} \leftarrow \theta$;
**for** $batch \in d_{train}$ **do**
  $\mathcal{L}_{batch} \leftarrow 0$;
  $L_{present} \leftarrow \{\, l \in L \mid l \in batch \,\}$ ; /* Languages present in batch */
  $scaler \leftarrow \sum_{l \in L_{present}} e^{q_l}$;
  **for** $l \in L_{present}$ **do**
    $q'_l \leftarrow \frac{e^{q_l}}{scaler}$;
    $batch_l \leftarrow \{\, pnt \in batch \mid pnt \in l \,\}$ ; /* Data points of language l in batch */
    $\mathcal{L}_{batch,l} \leftarrow \frac{\sum_{pnt \in batch_l} \mathcal{L}_{pnt}}{|batch_l|}$;
    $\mathcal{L}_{batch} \leftarrow \mathcal{L}_{batch} + q'_l \mathcal{L}_{batch,l}$;
  **end**
  $\mathcal{L}_{batch}.backward()$ ; /* Compute gradients */
  $\theta_{ft} \leftarrow opt.step(\theta_{ft})$ ; /* Update model parameters */
  **for** $l \in L_{present}$ **do**
    $q_l \leftarrow opt.step(q_l)$ ; /* Update group weights */
  **end**
  $sched.step()$ ; /* Update learning rate */
**end**

**Algorithm 5:** Fine-tuning with Worst Group Loss

**Input:**
$k\%$ - size (in percentage) of the worst group;
$\theta$ - base model;
$opt$ - optimizer;
$sched$ - learning rate scheduler;
$d_{train}$ - training data;
**Output:**
$\theta_{ft}$ - fine-tuned model with worst group loss;

---

$\theta_{ft} \leftarrow \theta$;
**for** $batch \in d_{train}$ **do**
  $size_g \leftarrow \lfloor |batch| * k\% \rfloor$;
  $L_{batch} \leftarrow []$;
  **for** $pnt \in batch$ **do**
    $L_{batch} \leftarrow L_{batch} + [\mathcal{L}_{pnt}]$ ; /* Append to list of loss values */
  **end**
  $L_{batch} \leftarrow sort\_in\_descending\_order(L_{batch})[: size_g]$ ; /* Select top loss values */
  $L_{batch} \leftarrow \frac{\sum_{\mathcal{L} \in L_{batch}} \mathcal{L}}{size_g}$
  $\mathcal{L}_{batch}.backward()$ ; /* Compute gradients */
  $\theta_{ft} \leftarrow opt.step(\theta_{ft})$ ; /* Update model parameters */
  $sched.step()$ ; /* Update learning rate */
**end**

| Language | Family |
|---|---|
| Afrikaans | Indo-European |
| Tosk Albanian | Indo-European |
| Amharic | Afro-Asiatic |
| Egyptian Arabic | Afro-Asiatic |
| South Azerbaijani | Turkic |
| Bambara | Mande |
| Bengali | Indo-European |
| Catalan | Indo-European |
| Welsh | Indo-European |
| Danish | Indo-European |
| German | Indo-European |
| Greek | Indo-European |
| Ewe | Atlantic-Congo |
| Finnish | Uralic |
| Fon | Atlantic-Congo |
| French | Indo-European |
| Hausa | Afro-Asiatic |
| Hebrew | Afro-Asiatic |
| Hindi | Indo-European |
| Croatian | Indo-European |
| Hungarian | Uralic |
| Armenian | Indo-European |
| Igbo | Atlantic-Congo |
| Indonesian | Austronesian |
| Icelandic | Indo-European |
| Italian | Indo-European |
| Javanese | Austronesian |
| Japanese | Japonic |
| Kannada | Dravidian |
| Georgian | Kartvelian |
| Halh Mongolian | Mongolic |
| Khmer | Austroasiatic |
| Kinyarwanda | Atlantic-Congo |
| Korean | Koreanic |
| Luganda | Atlantic-Congo |
| Luo | Nilotic |
| Standard Latvian | Indo-European |
| Malayalam | Dravidian |
| Mossi | Atlantic-Congo |
| Burmese | Sino-Tibetan |
| Dutch | Indo-European |
| Norwegian Bokmål | Indo-European |
| Chewa | Atlantic-Congo |
| Western Persian | Indo-European |
| Polish | Indo-European |
| Portuguese | Indo-European |
| Romanian | Indo-European |
| Russian | Indo-European |
| Slovak | Indo-European |
| Slovene | Indo-European |
| Shona | Atlantic-Congo |
| Spanish | Indo-European |
| Swedish | Indo-European |
| Swahili | Niger-Congo |
| Tamil | Dravidian |
| Telugu | Dravidian |
| Tagalog | Austronesian |
| Thai | Kra-Dai |
| Tswana | Atlantic-Congo |
| Turkish | Turkic |
| Twi | Atlantic-Congo |
| Urdu | Indo-European |
| Vietnamese | Austroasiatic |
| Wolof | Atlantic-Congo |
| Xhosa | Atlantic-Congo |
| Yoruba | Atlantic-Congo |
| Chinese | Sino-Tibetan |
| Standard Malay | Austronesian |
| Zulu | Atlantic-Congo |

Table 12: List of languages for bootstrapping synthetic projection labels.

### F.5 Adversarial Results

Due to language mismatch or translation error, correct span mappings might not exist for a particular source-target pair. Ideally, we would prefer a projection method to not produce any mapping in such scenarios. Quantitatively, we can measure a method's robustness in mismatch cases by constructing adversarial pairs where each source English sample is paired with a differently sampled text in the target language. To inspect, we conduct this experiment with the RELX-EXP dataset and report the projection rate (lower is better i.e. stronger rejection) of the baseline CODEC and our framework in Table 29. As shown in the table, while CODEC still attempts to project more than $50\%$ of the pairs, our framework successfully rejected adversarial pairs for all 4 languages, showing improved robustness.

| Dataset | Task |
|---|---|
| UNER (Mayhew et al., 2024) | Named Entity Recognition |
| MASSIVE (FitzGerald et al., 2023) | Slot Filling |
| MBABSA (Wu et al., 2025) | Aspect-based Sentiment Analysis |
| RELX (Koksal and Ozgur, 2020) | Relation Classification |

Table 13: List of datasets in the EXP benchmark

| Language | #Num. Samples |
|---|---|
| German | 208 |
| Portuguese | 77 |
| Swedish | 299 |
| Chinese | 228 |
| Total | 812 |

Table 14: Language-wise statistics for UNER-EXP

### F.6 Framework Ablations

We show ablation results in Table 30. As both EM and SM are recall-oriented, to measure precision, we additionally report the EXACT-PRECISION (EP) metric, calculated as follows:

$$EP = \frac{1}{n_{valid}} \sum_{v \in S} (\prod_{i=0}^{q_v} y^i_{tgt,v} \equiv l^i_{tgt,v})$$

where $n_{valid}$ denotes the number of samples with valid projection[18] (e.g. projection might be incorrect but the spans are valid).

Component-wise, adding the VERIFIER improves results on all 3 metrics, showcasing its usefulness in improving projection qualities. Meanwhile, PAIR AUGMENTATION (PA) does not improve recall-oriented scores (EM/SM) but instead improves precision-oriented scores (EP). This is within our expectations as PA was included to

[18] Invalid projection occurs in the event of *projection failure*, which we explain in Sec. G

| Language | #Num. Samples |
|---|---|
| Afrikaans | 1429 |
| Amharic | 1184 |
| Arabic | 1253 |
| Azerbaijani | 1309 |
| Bengali | 1152 |
| Catalan | 1304 |
| Welsh | 1505 |
| Danish | 1424 |
| German | 1515 |
| Greek | 1246 |
| Persian | 1177 |
| Finnish | 1430 |
| French | 1598 |
| Hebrew | 1304 |
| Hindi | 1221 |
| Hungarian | 1338 |
| Armenian | 1666 |
| Indonesian | 1347 |
| Icelandic | 1357 |
| Italian | 1276 |
| Javanese | 1320 |
| Japanese | 1395 |
| Kannada | 1414 |
| Georgian | 1412 |
| Khmer | 1245 |
| Korean | 1240 |
| Latvian | 1212 |
| Malayalam | 1180 |
| Mongolian | 1260 |
| Malay | 1229 |
| Burmese | 1226 |
| Dutch | 1522 |
| Norwegian | 1516 |
| Polish | 1438 |
| Portuguese | 1364 |
| Romanian | 1348 |
| Russian | 1248 |
| Slovene | 1254 |
| Spanish | 1422 |
| Albanian | 1120 |
| Swahili | 1229 |
| Swedish | 1394 |
| Tamil | 1243 |
| Telugu | 1265 |
| Tagalog | 1498 |
| Thai | 1235 |
| Turkish | 1350 |
| Urdu | 1211 |
| Vietnamese | 1341 |
| Chinese | 1224 |
| Total | 66390 |

Table 15: Language-wise statistics for MASSIVE-EXP

| Language | #Num. Samples |
|---|---|
| Arabic | 2065 |
| Danish | 2310 |
| German | 2264 |
| French | 2041 |
| Hindi | 2276 |
| Croatian | 2313 |
| Indonesian | 2193 |
| Japanese | 2200 |
| Korean | 2200 |
| Dutch | 2216 |
| Portuguese | 2238 |
| Russian | 2284 |
| Slovak | 2302 |
| Spanish | 2088 |
| Swahili | 2282 |
| Swedish | 2295 |
| Thai | 1987 |
| Turkish | 2186 |
| Vietnamese | 2059 |
| Chinese | 2228 |
| Total | 44027 |

Table 16: Language-wise statistics for MBABSA-EXP

| Language | #Num. Samples |
|---|---|
| German | 434 |
| French | 434 |
| Spanish | 434 |
| Turkish | 434 |
| Total | 1736 |

Table 17: Language-wise statistics for RELX-EXP

| Language | Retained Samples (%) | Retained Spans (%) | Retained length (#words) | All length (#words) |
|---|---|---|---|---|
| Chinese | 22.82 | 23.99 | 21.00 | 21.40 |
| German | 28.22 | 33.33 | 19.59 | 20.01 |
| Portuguese | 26.92 | 41.38 | 18.51 | 17.51 |
| Swedish | 29.96 | 35.41 | 18.98 | 19.06 |

Table 18: Effect of alignment heuristics on UNER

make the model more *careful* and generalize better to scenarios not naturally covered in training (e.g. target projection labels do not exist or projection is performed in new directions). As a result, the model recalls less but in exchange becomes more robust and makes fewer mistakes (as indicated in EP). Lastly, applying loss weighting (LGL/WGL) further improves model's performance, demonstrating that controlling point influence is essential for better results in projection training. Interestingly, WGL produces better results than LGL. We hypothesize that this might be because LGL's guidance is more generic (no constraint on the learnable weights), whereas WGL prioritizes the more difficult data points in each batch and thus provides

| Language | Retained Samples (%) | Retained Spans (%) | Retained length (#words) | All length (#words) |
|---|---|---|---|---|
| Afrikaans | 95.20 | 91.76 | 7.33 | 7.38 |
| Amharic | 95.48 | 91.85 | 5.81 | 5.94 |
| Arabic | 97.51 | 95.49 | 5.39 | 5.46 |
| Azerbaijani | 95.76 | 93.68 | 6.00 | 6.03 |
| Bengali | 97.38 | 94.98 | 6.31 | 6.39 |
| Catalan | 96.17 | 93.07 | 7.62 | 7.71 |
| Welsh | 95.86 | 92.57 | 8.29 | 8.38 |
| Danish | 96.80 | 94.18 | 6.77 | 6.84 |
| German | 95.16 | 91.28 | 6.87 | 6.95 |
| Greek | 95.19 | 91.52 | 7.31 | 7.37 |
| Spanish | 95.95 | 93.14 | 7.63 | 7.72 |
| Persian | 95.30 | 91.89 | 7.50 | 7.58 |
| Finnish | 95.84 | 92.99 | 4.97 | 5.02 |
| French | 95.40 | 92.55 | 8.10 | 8.19 |
| Hebrew | 95.60 | 93.76 | 5.81 | 5.90 |
| Hindi | 95.92 | 92.77 | 8.01 | 8.17 |
| Hungarian | 96.96 | 94.73 | 6.01 | 6.02 |
| Armenian | 94.82 | 90.75 | 5.99 | 6.10 |
| Indonesian | 96.01 | 93.40 | 6.71 | 6.78 |
| Icelandic | 95.56 | 92.49 | 6.70 | 6.76 |
| Italian | 96.08 | 93.16 | 7.00 | 7.12 |
| Japanese | 95.81 | 92.67 | 8.97 | 9.07 |
| Javanese | 95.17 | 91.43 | 6.33 | 6.39 |
| Georgian | 95.73 | 93.35 | 4.49 | 4.55 |
| Khmer | 92.63 | 90.80 | 3.33 | 3.34 |
| Kannada | 96.32 | 93.87 | 5.65 | 5.75 |
| Korean | 95.90 | 93.31 | 7.24 | 7.32 |
| Latvian | 95.51 | 92.19 | 5.73 | 5.81 |
| Malayalam | 95.86 | 93.45 | 5.50 | 5.61 |
| Mongolian | 96.04 | 94.54 | 6.26 | 6.33 |
| Malay | 96.32 | 93.55 | 6.82 | 6.91 |
| Burmese | 96.16 | 93.77 | 4.31 | 4.35 |
| Norwegian | 95.47 | 92.41 | 6.77 | 6.84 |
| Dutch | 95.30 | 91.62 | 7.44 | 7.50 |
| Polish | 96.25 | 93.20 | 6.07 | 6.15 |
| Portuguese | 94.59 | 91.14 | 7.55 | 7.68 |
| Romanian | 96.91 | 94.51 | 6.97 | 7.03 |
| Russian | 96.07 | 92.88 | 6.08 | 6.17 |
| Slovene | 96.02 | 93.23 | 6.06 | 6.16 |
| Albanian | 94.75 | 91.04 | 7.26 | 7.36 |
| Swedish | 96.07 | 93.16 | 6.57 | 6.66 |
| Swahili | 93.96 | 89.53 | 6.82 | 6.97 |
| Tamil | 96.13 | 93.60 | 5.47 | 5.56 |
| Telugu | 96.49 | 93.81 | 6.11 | 6.22 |
| Thai | 94.13 | 91.08 | 6.80 | 6.88 |
| Tagalog | 95.47 | 92.17 | 8.37 | 8.43 |
| Turkish | 95.74 | 92.86 | 5.78 | 5.85 |
| Urdu | 96.49 | 94.15 | 8.40 | 8.46 |
| Vietnamese | 95.99 | 92.89 | 9.20 | 9.29 |
| Chinese | 95.85 | 92.96 | 6.77 | 6.80 |

Table 19: Effect of alignment heuristics on MASSIVE

| Language | Retained Samples (%) | Retained Spans (%) | Retained length (#words) | All length (#words) |
|---|---|---|---|---|
| Arabic | 76.14 | 58.19 | 15.13 | 17.83 |
| Danish | 84.90 | 65.86 | 17.51 | 20.24 |
| German | 83.02 | 64.16 | 18.02 | 20.91 |
| Spanish | 77.59 | 60.90 | 18.53 | 21.55 |
| French | 81.28 | 63.27 | 18.51 | 21.38 |
| Hindi | 83.28 | 64.53 | 19.05 | 22.00 |
| Croatian | 84.97 | 66.07 | 16.27 | 18.78 |
| Indonesian | 81.04 | 62.66 | 16.00 | 18.89 |
| Japanese | 80.56 | 62.37 | 23.91 | 28.16 |
| Korean | 80.56 | 61.61 | 21.72 | 25.40 |
| Dutch | 82.17 | 63.98 | 18.07 | 21.03 |
| Portuguese | 82.83 | 64.70 | 17.81 | 20.51 |
| Russian | 83.76 | 65.07 | 16.65 | 19.32 |
| Slovak | 84.51 | 65.56 | 16.26 | 18.78 |
| Swedish | 84.28 | 65.37 | 16.84 | 19.41 |
| Swahili | 83.93 | 64.95 | 16.04 | 18.61 |
| Thai | 74.20 | 56.52 | 16.06 | 19.35 |
| Turkish | 80.81 | 62.12 | 14.22 | 16.63 |
| Vietnamese | 77.35 | 60.03 | 22.35 | 26.44 |
| Chinese | 81.58 | 62.36 | 17.18 | 20.25 |

Table 20: Effect of alignment heuristics on MBABSA

| Metric | Awesome | WSP | Binary-U | Binary-W | Codec | PVP |
|---|---|---|---|---|---|---|
| EM | 79.82 | 82.86 | 88.35 | 88.51 | 88.61 | **93.55** |
| SM | 84.36 | 87.78 | 91.77 | 91.82 | 94.22 | **96.93** |

Table 21: Average EM/SM scores of each method over all datasets in EXP.

| Language | Awesome | WSP | Binary-U | Binary-W | Codec | PVP |
|---|---|---|---|---|---|---|
| German | 93.80 | 95.64 | 93.57 | 93.94 | 95.71 | **96.52** |
| Portuguese | 94.10 | 97.40 | 97.40 | 97.4 | 99.18 | **99.44** |
| Swedish | 96.84 | 95.15 | 97.67 | **97.7** | 95.30 | **97.7** |
| Chinese | 82.20 | 82.47 | 83.53 | 84.08 | 89.00 | **94.61** |
| Average | 91.73 | 92.67 | 93.04 | 93.28 | 94.80 | **97.07** |

Table 22: SM scores on UNER-EXP. Best score on each row is highlighted.

| Language | Awesome | WSP | Binary-U | Binary-W | Codec | PVP |
|---|---|---|---|---|---|---|
| Afrikaans | 86.52 | 94.43 | 93.07 | 93.15 | 96.38 | **97.65** |
| Amharic | 9.52 | 18.92 | 90.87 | 90.99 | 87.67 | **92.7** |
| Arabic | 77.51 | 81.32 | 89.57 | 89.12 | 88.41 | **93.17** |
| Azerbaijani | 65.91 | 80.57 | 90.08 | 89.97 | 82.42 | **93.33** |
| Bengali | 70.78 | 82.31 | 94.49 | 95.15 | 92.1 | **96.09** |
| Catalan | 73.59 | 75.96 | 76.83 | 75.73 | 92.69 | **95.55** |
| Welsh | 58.47 | 68.76 | 71.36 | 71.62 | **92.09** | 86.37 |
| Danish | 88.34 | 93.67 | 95.83 | 95.70 | 97.54 | **98.68** |
| German | 91.66 | 92.42 | 93.77 | 93.53 | 95.64 | **97.51** |
| Greek | 78.10 | 86.59 | 85.4 | 85.57 | 94.61 | **97.28** |
| Persian | 71.89 | 82.05 | 93.32 | 93.59 | 91.43 | **96.23** |
| Finnish | 83.07 | 88.33 | 95.32 | 95.26 | 97.01 | **97.53** |
| French | 78.99 | 82.80 | 82.17 | 81.99 | 93.39 | **95.5** |
| Hebrew | 76.48 | 82.87 | 93.03 | 92.86 | 90.98 | **95.41** |
| Hindi | 73.05 | 79.22 | 87.8 | 87.65 | 90.43 | **94.1** |
| Hungarian | 78.00 | 85.05 | 90.11 | 90.32 | 93.04 | **96.68** |
| Armenian | 76.17 | 82.30 | 94.14 | 94.07 | 88.42 | **95.95** |
| Indonesian | 69.43 | 81.65 | 90.41 | 90.24 | 92.44 | **94.33** |
| Icelandic | 74.60 | 81.53 | 88.78 | 88.39 | 94.52 | **96.48** |
| Italian | 78.32 | 81.68 | 83.61 | 83.22 | 92.03 | **93.91** |
| Javanese | 57.01 | 65.82 | 83.4 | 82.47 | **92.52** | 91.09 |
| Japanese | 70.77 | 71.74 | 84.19 | 84.53 | 64.81 | **92.9** |
| Kannada | 69.61 | 80.52 | 90.56 | 90.57 | 89.28 | **92.07** |
| Georgian | 80.78 | 81.95 | **96.89** | 96.42 | 92.62 | 96.79 |
| Khmer | 40.92 | 34.62 | 95.5 | 96.40 | 92.15 | **96.77** |
| Korean | 65.36 | 74.42 | 85.3 | 84.86 | 79.25 | **87.57** |
| Latvian | 79.40 | 84.23 | 94.44 | 94.41 | 94.79 | **98.63** |
| Malayalam | 69.89 | 77.77 | 94.57 | 95.06 | 94.04 | **95.76** |
| Mongolian | 45.75 | 66.20 | 89.54 | 89.64 | 86.02 | **91.32** |
| Malay | 67.40 | 81.65 | 92.61 | 92.03 | 93.66 | **97.06** |
| Burmese | 69.73 | 78.27 | 95.93 | 97.27 | 93.39 | **97.32** |
| Dutch | 88.08 | 91.71 | 89.89 | 89.16 | 95.29 | **96.78** |
| Norwegian | 86.80 | 92.74 | 94.04 | 93.44 | 96.15 | **97.38** |
| Polish | 83.66 | 86.01 | 90.44 | 90.43 | 93.26 | **95.6** |
| Portuguese | 72.61 | 81.17 | 78.3 | 77.38 | 93.69 | **95.0** |
| Romanian | 78.50 | 78.10 | 83.3 | 82.43 | 94.48 | **96.22** |
| Russian | 88.24 | 92.34 | 92.51 | 92.39 | 95.74 | **97.27** |
| Slovene | 81.46 | 86.37 | 91.78 | 91.47 | 94.79 | **96.82** |
| Spanish | 74.79 | 81.73 | 77.02 | 76.96 | 93.67 | **95.49** |
| Albanian | 68.98 | 77.08 | 78.22 | 77.35 | 92.69 | **94.23** |
| Swahili | 44.29 | 48.62 | 67.79 | 66.58 | **90.21** | 87.48 |
| Swedish | 88.51 | 92.06 | 94.13 | 93.73 | 96.13 | **98.32** |
| Tamil | 71.35 | 81.46 | 92.08 | 91.86 | 91.48 | **93.51** |
| Telugu | 67.66 | 79.95 | 93.56 | 93.86 | 91.49 | **95.43** |
| Tagalog | 63.00 | 71.58 | 73.82 | 73.41 | **94.74** | 91.02 |
| Thai | 56.62 | 35.78 | 87.75 | 86.99 | 81.82 | **92.32** |
| Turkish | 66.90 | 76.67 | 93.87 | 94.30 | 89.15 | **95.71** |
| Urdu | 60.71 | 71.75 | 84.16 | 84.27 | 89.93 | **91.99** |
| Vietnamese | 47.52 | 74.62 | 83.68 | 81.68 | 79.38 | **91.1** |
| Chinese | 82.34 | 81.76 | 91.56 | 90.79 | 70.38 | **95.58** |
| Average | 70.98 | 77.62 | 88.42 | 88.21 | 90.8 | **94.78** |

Table 23: SM scores on MASSIVE-EXP. Best score on each row is highlighted.

more specific guidance, which leads to better training.

## G Projection failure

For measuring projection rate (i.e. percentage of successfully projected samples), we rely on the notion of *projection failure* - cases where a projection method cannot produce any valid span mapping for a given input sample. Intuitively, this is analogous to a model that produces no prediction at all, which is different from producing an incorrect prediction. In terms of main metric computation (EM and SM), this distinction is only relevant for formal completeness and does not affect scores.

| Language | Awesome | WSP | Binary-U | Binary-W | Codec | PVP |
|---|---|---|---|---|---|---|
| Arabic | 82.03 | 89.28 | 93.11 | 93.30 | 94.26 | **97.13** |
| Danish | 96.60 | 97.86 | 99.11 | 99.05 | 98.83 | **99.71** |
| German | 93.21 | 94.08 | 95.32 | 95.45 | 97.14 | **98.6** |
| French | 82.64 | 87.00 | 88.70 | 88.75 | 95.06 | **96.02** |
| Hindi | 87.00 | 92.84 | 96.46 | 96.42 | 98.32 | **98.94** |
| Croatian | 89.70 | 93.46 | 94.01 | 93.99 | 97.78 | **98.96** |
| Indonesian | 88.03 | 93.19 | 95.58 | 95.70 | 96.62 | **98.81** |
| Japanese | 81.59 | 71.04 | 87.71 | 87.88 | 89.93 | **96.25** |
| Korean | 76.87 | 82.84 | 88.36 | 90.48 | 85.57 | **93.96** |
| Dutch | 93.76 | 95.22 | 95.46 | 95.69 | 97.89 | **98.7** |
| Portuguese | 82.43 | 86.05 | 86.36 | 86.03 | 97.6 | **97.89** |
| Russian | 89.62 | 92.88 | 93.78 | 93.38 | 97.51 | **98.66** |
| Slovak | 91.79 | 94.28 | 95.78 | 95.83 | 98.34 | **98.38** |
| Spanish | 79.34 | 83.17 | 83.37 | 83.40 | 94.32 | **95.46** |
| Swahili | 66.13 | 75.26 | 79.12 | 78.52 | **98.22** | 97.02 |
| Swedish | 96.44 | 96.84 | 98.96 | 98.96 | 98.76 | **99.62** |
| Thai | 61.56 | 43.54 | 87.88 | 88.08 | 89.26 | **92.94** |
| Turkish | 84.18 | 90.19 | 96.50 | 96.53 | 96.56 | **97.93** |
| Vietnamese | 52.28 | 87.67 | 90.48 | 90.03 | 93.39 | **95.31** |
| Chinese | 88.40 | 91.35 | 94.30 | 94.14 | 93.23 | **98.09** |
| Average | 83.18 | 86.90 | 92.02 | 92.08 | 95.43 | **97.42** |

Table 24: SM scores on MBABSA-EXP. Best score on each row is highlighted.

| Language | Awesome | WSP | Binary-U | Binary-W | Codec | PVP |
|---|---|---|---|---|---|---|
| German | 96.10 | 96.35 | 96.45 | 96.58 | 97.25 | **98.22** |
| French | 94.51 | 93.88 | 94.09 | 94.04 | 96.22 | **98.03** |
| Spanish | 86.82 | 89.23 | 87.65 | 87.42 | 94.42 | **98.7** |
| Turkish | 88.75 | 96.25 | 96.19 | 96.77 | 95.54 | **98.79** |
| Average | 91.55 | 93.93 | 93.59 | 93.70 | 95.86 | **98.44** |

Table 25: SM scores on RELX-EXP. Best score on each row is highlighted.

Since EM and SM require comparing predicted spans with ground-truth spans, we treat the predicted span set as empty when projection failure occurs. As a result, the score for that instance is zero (assuming reference label is non-empty, which holds for EXP).

## H Additional Benchmarking Details

**Baselines** For AWESOME, we use the `model_without_co` checkpoint released by (Dou and Neubig, 2021), with features from layer 8 and threshold value $1e-3$, following their settings. For WSP, we use their released `WSPAlign-ft-kftt` checkpoint (Wu et al., 2023), which we found to achieve the best performance. For CODEC (Le et al., 2024), we use their released fine-tuned checkpoint (`600M`) and followed them in setting up corresponding hyperparameters. For speed benchmarking, we set the batch size of AWESOME, WSP, BINARY and PVP to 16. For CODEC which involves multiple inference phases, we set the batch size of stage 3 and 4 to 8 and 256, respectively.

**LLMs** For MIPROV2, we set the number of candidate instructions and demonstration sets as $m = 16$. In addition, we apply minibatching with batch size $||\mathcal{B}|| = 25$ and mini evaluation steps as $f = 10$. To sample instruction proposals, we set temperate $T_{init} = 0.5$. For Bayesian optimization, we set the maximum number of iterations as 50. The in-context exemplar candidates are

| Language | PVP | Codec | MIPROv2 | CLaP | SFT-small | SFT |
|---|---|---|---|---|---|---|
| Afrikaans | **97.65** | 96.38 | 81.81 | 65.64 | 94.27 | 96.94 |
| Amharic | **92.7** | 87.67 | 28.33 | 59.22 | 62.88 | 76.72 |
| Arabic | 93.17 | 88.41 | 67.80 | 69.47 | 89.30 | **94.3** |
| Azerbaijani | **93.33** | 82.42 | 27.21 | 79.61 | 69.65 | 89.06 |
| Bengali | **96.09** | 92.1 | 44.48 | 78.32 | 86.04 | **96.09** |
| Catalan | **95.55** | 92.69 | 60.81 | 70.43 | 84.20 | 90.85 |
| Welsh | 86.37 | **92.09** | 61.28 | 63.90 | 68.11 | 87.1 |
| Danish | **98.68** | 97.54 | 83.00 | 83.26 | 95.26 | 97.36 |
| German | 97.51 | 95.64 | 80.62 | 78.09 | 95.41 | **97.87** |
| Greek | **97.28** | 94.61 | 77.12 | 72.72 | 86.82 | 95.1 |
| Persian | **96.23** | 91.43 | 46.29 | 84.16 | 81.84 | 94.66 |
| Finnish | 97.53 | 97.01 | 68.78 | 83.10 | 85.85 | **98.26** |
| French | **95.5** | 93.39 | 69.74 | 60.23 | 90.08 | 93.01 |
| Hebrew | **95.41** | 90.98 | 65.80 | 81.57 | 88.07 | 93.83 |
| Hindi | **94.1** | 90.43 | 45.98 | 78.54 | 82.05 | 92.05 |
| Hungarian | **96.68** | 93.04 | 64.86 | 75.54 | 78.05 | 95.35 |
| Armenian | **95.95** | 88.42 | 55.94 | 83.95 | 72.68 | 91.14 |
| Indonesian | **94.33** | 92.44 | 61.27 | 80.02 | 90.53 | 93.59 |
| Icelandic | **96.48** | 94.52 | 69.19 | 77.28 | 82.69 | 92.29 |
| Italian | **93.91** | 92.03 | 67.98 | 72.45 | 89.51 | 92.52 |
| Javanese | 91.09 | **92.52** | 59.75 | 70.23 | 77.26 | 87.32 |
| Japanese | **92.9** | 64.81 | 32.04 | 80.44 | 88.19 | 90.14 |
| Kannada | **92.07** | 89.28 | 45.00 | 78.54 | 73.75 | 90.04 |
| Georgian | **96.79** | 92.62 | 62.60 | 82.59 | 78.60 | 94.2 |
| Khmer | **96.77** | 92.15 | 46.95 | 82.19 | 71.82 | 93.31 |
| Korean | 87.57 | 79.25 | 34.59 | 77.18 | 85.60 | **89.98** |
| Latvian | **98.63** | 94.79 | 68.44 | 82.29 | 82.23 | 96.97 |
| Malayalam | **95.76** | 94.04 | 38.45 | 76.60 | 78.44 | 92.62 |
| Mongolian | **91.32** | 86.02 | 23.29 | 73.13 | 62.93 | 75.62 |
| Malay | **97.06** | 93.66 | 58.34 | 85.38 | 89.09 | 95.99 |
| Burmese | **97.32** | 93.39 | 24.19 | 85.69 | 72.74 | 89.99 |
| Dutch | **96.78** | 95.29 | 80.49 | 75.52 | 93.06 | 94.73 |
| Norwegian | **97.38** | 96.15 | 79.12 | 78.93 | 93.92 | 97.13 |
| Polish | **95.6** | 93.26 | 69.40 | 75.83 | 90.67 | 93.76 |
| Portuguese | **95.0** | 93.69 | 68.02 | 72.98 | 90.07 | 91.75 |
| Romanian | **96.22** | 94.48 | 66.89 | 78.31 | 82.87 | 93.42 |
| Russian | **97.27** | 95.74 | 75.16 | 79.46 | 94.99 | 96.45 |
| Slovene | **96.82** | 94.79 | 66.74 | 79.38 | 86.50 | 95.66 |
| Spanish | **95.49** | 93.67 | 67.28 | 72.80 | 90.85 | 90.71 |
| Albanian | **94.23** | 92.69 | 63.87 | 74.63 | 72.94 | 87.97 |
| Swahili | 87.48 | **90.21** | 55.43 | 75.40 | 60.59 | 75.12 |
| Swedish | **98.32** | 96.13 | 82.63 | 81.70 | 93.99 | 96.16 |
| Tamil | **93.51** | 91.48 | 40.52 | 76.16 | 71.96 | 89.31 |
| Telugu | **95.43** | 91.49 | 48.30 | 81.78 | 76.55 | 92.73 |
| Tagalog | 91.02 | **94.74** | 65.22 | 73.81 | 74.02 | 89.52 |
| Thai | 92.32 | 81.82 | 55.41 | 82.32 | 90.00 | **93.0** |
| Turkish | **95.71** | 89.15 | 22.97 | 77.31 | 82.26 | 93.2 |
| Urdu | **91.99** | 89.93 | 40.03 | 77.78 | 67.68 | 87.09 |
| Vietnamese | 91.1 | 79.38 | 35.89 | 71.47 | 84.92 | **91.11** |
| Chinese | **95.58** | 70.38 | 49.89 | 85.15 | 92.22 | 93.76 |
| Average | **94.78** | 90.8 | 57.10 | 76.85 | 82.48 | 91.94 |

Table 26: SM scores (including LLMs) on MASSIVE-EXP. Best score on each row is highlighted.

| Language | PVP | Codec | MIPROv2 | CLaP | SFT-small | SFT |
|---|---|---|---|---|---|---|
| German | **96.52** | 95.71 | 81.19 | 81.44 | 96.45 | 96.35 |
| Portuguese | 99.44 | 99.18 | 78.60 | 83.85 | 99.89 | **100.0** |
| Swedish | **97.7** | 95.30 | 84.58 | 90.21 | 97.54 | **97.7** |
| Chinese | 94.61 | 89.00 | 69.58 | 90.43 | 94.37 | **95.36** |
| Average | 97.07 | 94.80 | 78.49 | 86.48 | 97.06 | **97.35** |

Table 27: SM scores (including LLMs) on UNER-EXP. Best score on each row is highlighted.

| Language | Codec | PVP | English |
|---|---|---|---|
| Afrikaans | 66.2 | **68.9** | 56.3 |
| Amharic | 56.6 | **57.3** | 30.8 |
| Egyptian Arabic | 57.2 | **61.1** | 45.4 |
| South Azerbaijani | 40.0 | 38.0 | **43.5** |
| Bengali | 66.1 | **66.4** | 46.3 |
| Catalan | 61.8 | **64.5** | 52.3 |
| Welsh | 53.4 | **56.5** | 31.8 |
| Danish | 70.5 | **72.1** | 62.6 |
| German | 65.0 | **67.6** | 60.0 |
| Greek | 62.9 | **66.8** | 56.7 |
| Western Persian | 66.4 | **70.6** | 59.8 |
| Finnish | 63.9 | **67.1** | 60.2 |
| French | 64.6 | **69.3** | 60.7 |
| Hebrew | 60.3 | **64.0** | 44.8 |
| Hindi | **65.2** | 65.0 | 51.8 |
| Hungarian | 62.8 | **66.6** | 53.9 |
| Armenian | 56.7 | **60.8** | 46.2 |
| Indonesian | 63.0 | **66.5** | 57.3 |
| Icelandic | 57.7 | **60.4** | 43.2 |
| Italian | 67.3 | **67.7** | 58.5 |
| Average | 61.38 | **63.86** | 51.1 |

Table 28: F1 scores of 20 languages in MASSIVE (downstream performance with XLM-ROBERTA-BASE). Best score on each row is highlighted.

| Language | Codec | PVP (ours) |
|---|---|---|
| German | 58.76 | **0.0** |
| French | 53.23 | **0.0** |
| Spanish | 56.22 | **0.0** |
| Turkish | 71.20 | **0.0** |
| Average | 59.85 | **0.0** |

Table 29: Adversarial Projection Rate (lower is better) on RELX-EXP

randomly bootstrapped from the synthetic training data of PVP. For CLAP, we used exemplars from the training splits of MASSIVE-EXP and MBABSA-EXP for their respective evaluations. On UNER-EXP and RELX-EXP where there is no training split, we used exemplars of the same language from MASSIVE-EXP. Note that these exemplars are solely used for the CLAP baseline as we observed that the method was very sensitive to exemplar quality, thus decided to sample from the training split of EXP. These exemplars were not exposed to any other baselines (including PVP). The number of exemplars is set at 3 for CLAP.

For SFT/SFT-SMALL, we use LORA with $r = 64$, $\alpha = 128$, $dropout = 0.05$, and insert adapters on all linear modules. Besides, we use mixed precision training with BFLOAT16. Each model is trained with an accumulated batch size of 32, with initial learning rate $5e-5$ and a linear learning rate scheduler without warmup and scheduler length set at 1000000. Checkpoints are evaluated every 250 steps and the checkpoint with the lowest development loss is chosen for test time inference. For optimizer, we use ADAMW (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-08$. We set the maximum training steps as 80000.

**Hyperparameter choice** For PVP, we did not exhaustively tune the hyperparameters as iterative re-training is expensive. For $k$, we use average of group number (one bad and one good groups), hence 50%. For $p_b$, we assume a uniform prior between two directions and thus set it as 0.5. For $p_n$, as the replaced samples could negatively impact model's recalls, we performed grid search in the range $\{0.2, 0.4, 0.6, 0.8\}$, finding 0.2 to achieve best results on the dev split. Still, we find dev EM scores to be rather stable across sweeps, suggesting that the model is not overly sensitive to $p_n$.

**Implicit Evaluation** For downstream training, we set the number of epochs as 5 and batch size as 32. Checkpoints are evaluated every epoch and the checkpoint with the highest development F1 score is chosen for test time inference. We use projected training data in the target language for training and ENGLISH development data for checkpoint selection of each method (except for the ENGLISH-FT baseline which instead uses ENGLISH data for training). For optimizer, we also use ADAMW (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-08$.

**Libraries** We made use of the TRANSFORMERS (Wolf et al., 2020) and PYTORCH (Paszke et al., 2019) libraries for implementations. For LLM baselines, we also utilize the VLLM (Kwon et al., 2023), MS-SWIFT (Zhao et al., 2024) and DSPY (Khattab et al., 2024) frameworks.

**Hardware** Experiments were conducted on 1 A100 PCIE GPU.

## I Licenses

We list the licenses of datasets used in the experiments.

- CONLL-2003: described in https://www.clips.uantwerpen.be/conll2003/ner/
- MASSIVE: cc-by-4.0
- MBABSA: apache-2.0

| Metric | Proposer | Proposer+Verifier | Proposer+Verifier+PA | Proposer+Verifier+PA+LGL | Proposer+Verifier+PA+WGL |
|---|---|---|---|---|---|
| EM | 92.53 | 92.92 | 92.90 | 93.00 | **93.55** |
| SM | 96.49 | 96.62 | 96.56 | 96.60 | **96.93** |
| EP | 93.00 | 93.33 | 93.53 | 93.60 | **93.95** |

Table 30: Ablations of each component in the framework. Scores are averaged on all test sets in EXP.

- RELX: MIT
- UNER: cc by-sa
- FINDVEHICLE: We could not find the license for this dataset
- ONTONOTES5: described in https://catalog.ldc.upenn.edu/LDC2013T19
- HARVEYNER: We could not find the license for this dataset
- CROSSNER: MIT