Improving Bilingual Lexicon Induction with Cross-Encoder Reranking

Yaoyiran Li, Fangyu Liu, Ivan Vulić[*] and **Anna Korhonen**[*]
Language Technology Lab, TAL, University of Cambridge
{yl711,fl399,iv250,alk23}@cam.ac.uk

Abstract

Bilingual lexicon induction (BLI) with limited bilingual supervision is a crucial yet challenging task in multilingual NLP. Current state-of-the-art BLI methods rely on the induction of cross-lingual word embeddings (CLWEs) to capture cross-lingual word similarities; such CLWEs are obtained **1)** via traditional static models (e.g., VECMAP), or **2)** by extracting type-level CLWEs from multilingual pretrained language models (mPLMs), or **3)** through combining the former two options. In this work, we propose a novel semi-supervised *post-hoc* reranking method termed **BLICEr** (**BLI** with **C**ross-**E**ncoder **R**eranking), applicable to any precalculated CLWE space, which improves their BLI capability. The key idea is to 'extract' cross-lingual lexical knowledge from mPLMs, and then combine it with the original CLWEs. This crucial step is done via **1)** creating a word similarity dataset, comprising positive word pairs (i.e., true translations) and hard negative pairs induced from the original CLWE space, and then **2)** fine-tuning an mPLM (e.g., mBERT or XLM-R) in a cross-encoder manner to predict the similarity scores. At inference, we **3)** combine the similarity score from the original CLWE space with the score from the BLI-tuned cross-encoder. BLICEr establishes new state-of-the-art results on two standard BLI benchmarks spanning a wide spectrum of diverse languages: it substantially outperforms a series of strong baselines across the board. We also validate the robustness of BLICEr with different CLWEs.

1 Introduction and Motivation

Bilingual lexicon induction (BLI) or word translation is one of the core tasks in multilingual NLP (Rapp, 1995; Gaussier et al., 2004; Shi et al., 2021; Li et al., 2022, *inter alia*), with its applications spanning machine translation (Qi et al., 2018; Duan et al., 2020), language acquisition and learning (Yuan et al., 2020), as well as supporting NLP tasks in low-resource scenarios (Heyman et al., 2018; Wang et al., 2022), among others. The predominant approach to BLI is based on the induction of a shared *cross-lingual word embedding* (CLWE) semantic space: word translation is then tackled by searching for the nearest neighbour in the other language. Recent BLI work has largely focused on the so-called *mapping-based* or *projection-based* methods (Mikolov et al., 2013; Ruder et al., 2019). Their prime advantage is strong performance coupled with largely reduced bilingual supervision, typically spanning only 1k-5k seed word pairs (Glavaš et al., 2019). This makes them fitting for weakly supervised setups and low-resource languages.

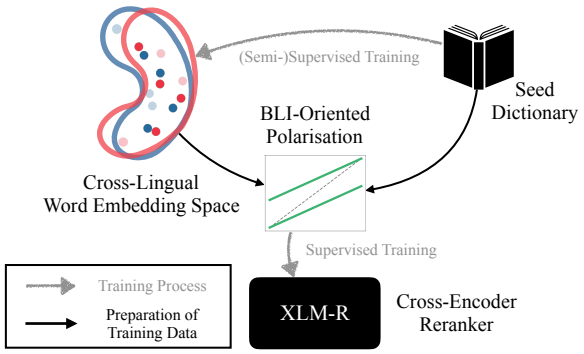

Figure 1: An overview of the proposed BLICEr approach, described in detail in §3.

In parallel, *cross-encoders* (CEs) have gained popularity in sentence-level NLP tasks that involve pairwise sentence comparisons. Unlike the so-called *embedding-based models* (also called bi-encoders or dual-encoders) which process two sequences independently and in parallel to create their embeddings and only model their late interaction (Henderson et al., 2020), CEs take the concatenation of two sequences as input and directly predict the similarity of the two sequences (Humeau et al., 2020). As the self-attention heads in CEs can simultaneously attend to tokens from both sequences, CEs are considered more power-

[*] Equal senior contribution.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4129 - 4145
December 7-11, 2022

ful sequence-pair models than embedding-based models which can only perform *post-hoc* comparisons in the embedding space. A large volume of evidence suggests that, under the same amount of supervision, CEs typically substantially outperform embedding-based models in information retrieval (Qu et al., 2021), dialogue (Urbanek et al., 2019), and semantic similarity tasks (Thakur et al., 2021; Liu et al., 2022), and their benefits are especially pronounced in low-data regimes with limited task supervision (Geigle et al., 2022).

Motivated by the work on CEs in sentence-level tasks, in this work, we propose to use CEs to benefit BLI. In a nutshell, we aim to expose useful word translation knowledge from multilingual pretrained language models (mPLMs) such as mBERT or XLM-R via BLI-oriented CE fine-tuning; this knowledge complements the knowledge stored in the CLWEs. We demonstrate that CEs can be effectively leveraged with cross-lingual word pairs, learning more accurate cross-lingual word similarity scores required for BLI. We present **BLICEr** (**BLI** with **C**ross-**E**ncoder **R**eranking), a *post-hoc* reranking method for BLI, illustrated in Figure 1 and described in detail in §3. BLICEr requires no additional supervision beyond the seed dictionary used for inducing (i.e., mapping) the CLWE space, and it can be combined with any existing CLWE approach, boosting their BLI performance.

We conduct extensive BLI experiments on two standard BLI benchmarks spanning a diverse language sample, covering 44 translation directions and a total of 352 different BLI setups. We observe large and consistent improvements brought about by BLICEr across the board: we report gains in 351 out of the 352 BLI setups over the very recent and strong BLI baseline of Li et al. (2022), establishing new state-of-the-art (SotA) performance. We also empirically validate that BLICEr is universally useful, yielding gains with different 'CLWE backbones', and run a series of insightful ablations to verify the usefulness of individual components involved in the BLICEr design. Our code is publicly available at `https://github.com/cambridgeltl/BLICEr`.

2 Related Work

BLI and CLWEs. The predominant BLI methods depend on learning linear or non-linear functions that map monolingual word embeddings to a shared CLWE space (Xing et al., 2015; Lample et al., 2018; Joulin et al., 2018; Artetxe et al., 2018; Grave et al., 2019; Patra et al., 2019; Jawanpuria et al., 2019; Mohiuddin et al., 2020; Glavaš and Vulić, 2020; Peng et al., 2021; Sachidananda et al., 2021). There have also been attempts to conduct BLI via monolingual and multilingual pretrained language models (Gonen et al., 2020; Vulić et al., 2020a,b, 2021). However, empirical evidence suggests that these approaches underperform static CLWEs for BLI (Vulić et al., 2020b): this is possibly because PLMs are primarily designed for longer sequence-level tasks and thus may naturally have inferior performance in word-level tasks when used off-the-shelf (Vulić et al., 2022). Recent work started to combine static and contextualised word representations for BLI (Zhang et al., 2021). In fact, the previous SotA CLWEs for BLI, used as the baseline model in our work, are derived via a two-stage contrastive learning approach combining word representations of both types (Li et al., 2022). Our work builds upon existing CLWE-based BLI methods, and proposes a novel *post-hoc* reranking method that universally enhances BLI performance of any backbone CLWE method.

Cross-Encoders. They have wide applications in text matching (Chen et al., 2020), semantic textual similarity (Thakur et al., 2021; Liu et al., 2022), and cross-modal retrieval (Geigle et al., 2022). They typically outperform the class of Bi-encoder models (Reimers and Gurevych, 2019), but are much more time-consuming and even prohibitively expensive to run for retrieval tasks directly (Geigle et al., 2022). While mPLMs as bi-encoders for BLI have been studied in very recent research (Li et al., 2022), to the best of our knowledge, BLICEr is the first work to leverage CEs for the BLI task.

3 Methodology

3.1 Background

BLI Task Description. We assume two languages, L_x (source) and L_y (target), with their respective vocabularies $\mathcal{X}=\{w_1^x,\ldots,w_{|\mathcal{X}|}^x\}$ and $\mathcal{Y}=\{w_1^y,\ldots,w_{|\mathcal{Y}|}^y\}$. Let us denote the Cartesian set of all possible cross-lingual word pairs as $\mathbf{\Pi} = \mathcal{X}\times\mathcal{Y}$, and a word pair from $\mathbf{\Pi}$ as $\pi=(w^x,w^y)$. As in a large body of recent BLI work that focuses on mapping-based BLI methods (Mikolov et al., 2013; Glavaš et al., 2019; Li et al., 2022, *inter alia*), we assume (i) $\mathcal{D}_S$, a set of seed word translation pairs for training, and (ii) $\mathcal{D}_T$, a test set of

word pairs for evaluation, such that $\mathcal{D}_S, \mathcal{D}_T \subset \mathbf{\Pi}$ and $\mathcal{D}_S \cap \mathcal{D}_T = \emptyset$. Similar to prior work, we then formulate the BLI task as learning a mapping function $f : \mathbf{\Pi} \rightarrow \mathbb{R}$; f in fact measures cross-lingual word similarity between the words from the input pair π (Heyman et al., 2017; Karan et al., 2020). At inference, the BLI task in the $L_x \rightarrow L_y$ translation direction is then to retrieve the most similar L_y word for each L_x word w^x in $\mathcal{D}_T$: this is the word $\hat{w}^y$ from $\mathcal{Y}$ that maximises the cross-lingual similarity score obtained by f. More formally:[1] $\hat{w}^y = \text{argmax}_{w^y \in \mathcal{Y}} f(w^x, w^y)$. In low-resource setups where only a small seed dictionary $\mathcal{D}_S$ is available as bilingual supervision, most state-of-the-art BLI approaches are still based on the induction of CLWEs (Ruder et al., 2019).

CLWEs. Let $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{X}| \times d}$ and $\boldsymbol{Y} \in \mathbb{R}^{|\mathcal{Y}| \times d}$ denote an already aligned (or shared) CLWE semantic space, where L_x words are represented by real-valued vectors/CLWEs from $\boldsymbol{X}$, and the representations of L_y words are provided in $\boldsymbol{Y}$. In other words, the d-dimensional row vector $\mathbf{x}_i$ from $\boldsymbol{X}$ correspond to the specific word $w_i^x \in \mathcal{X}$, and the same holds for the target language.

A plethora of different methods with various data requirements and bilingual supervision can be used to induce such CLWEs (Ruder et al., 2019). Most commonly, due to reduced bilingual supervision requirements, the CLWEs are induced by (i) pretraining monolingual word embeddings independently in two languages, and then (ii) mapping them by linear (Mikolov et al., 2013; Xing et al., 2015; Joulin et al., 2018; Artetxe et al., 2018) or non-linear transformations (Glavaš and Vulić, 2020; Mohiuddin et al., 2020), minimising the distance between the original monolingual word embedding spaces. Optionally, such static CLWEs can be combined or enhanced with external word-level knowledge such as word translation knowledge embedded in multilingual language models (Zhang et al., 2021; Li et al., 2022; Vulić et al., 2022).

The actual similarity function $f(\pi)$ is detached from the chosen CLWE method to obtain $\boldsymbol{X}$ and $\boldsymbol{Y}$. Following prior work (Li et al., 2022), $f(\pi)$ is the Cross-domain Similarity Local Scaling (CSLS) measure (Lample et al., 2018) between the associated embeddings $\mathbf{x} \in \boldsymbol{X}$ and $\mathbf{y} \in \boldsymbol{Y}$:

$$f_C(\pi) = \cos(\mathbf{x}, \mathbf{y}) - \Gamma_{\boldsymbol{X}}(\mathbf{y}) - \Gamma_{\boldsymbol{Y}}(\mathbf{x}). \quad (1)$$

Here, cos denotes the cosine similarity, $\Gamma_{\boldsymbol{X}}(\mathbf{y})$ is the average cosine similarity between $\mathbf{y}$ and its k nearest neighbours (typically $k = 10$) in $\boldsymbol{X}$; $\Gamma_{\boldsymbol{Y}}(\mathbf{x})$ is defined similarly. CSLS is a standard similarity function in BLI which typically outperforms the 'vanilla' cosine similarity, as it mitigates the hubness problem during inference.[2]

3.2 BLICEr: Cross-Encoder Reranking

Method in a Nutshell. The proposed BLICEr method is illustrated in Figure 1. The main idea is to refine the initial cross-lingual word similarity scores obtained from the original CLWE space (see Eq. 1). In particular, assuming the seed dictionary $\mathcal{D}_S$ and the precalculated CLWEs, we first derive positive translation pairs $\mathcal{D}_P \supseteq \mathcal{D}_S$ (true translation pairs) and hard negative pairs $\mathcal{D}_N$ (semantically similar words that do not constitute a real translation pair). We then *polarise* the scores for both $\mathcal{D}_P$ and $\mathcal{D}_N$ word pairs: the polarisation step effectively increases semantic similarity scores between positives and decreases them for negatives. We then use the polarised scores to fine-tune any mPLM (e.g., mBERT or XLM-R) to transform them into BLI-oriented cross-encoders: that is, we provide word pairs as input, aiming to predict the correct similarity score. Finally, the mPLMs, now transformed into BLI-focused cross-encoders, produce cross-lingual similarity scores for unseen word pairs, which work in synergy with and refine the similarity scores produced by the original cross-lingual word embeddings.

In what follows, we describe the main components of the full BLICEr post-processing method.

Constructing Sets of Positive and Negative Pairs. The cross-encoder fine-tuning crucially depends on the positive and negative pair sets $\mathcal{D}_P$ and $\mathcal{D}_N$. The construction of $\mathcal{D}_P$ starts from the set of gold translation pairs $\mathcal{D}_S$. Prior work demonstrated that additional highly reliable translation pairs can be extracted automatically from the CLWE space (Artetxe et al., 2018; Vulić et al., 2019). We thus follow the approach of Li et al. (2022), and extract additional N_{aug} high-confidence pairs $\mathcal{D}_{aug}$: they

[1] It is possible that a single word can have several plausible translations. Following previous work (Lample et al., 2018; Glavaš et al., 2019; Li et al., 2022), given a query word from the test set $\mathcal{D}_T$, as long as the model retrieves any of the ground-truth translations in its top K predictions, it is considered a correct prediction based on the standard Precision@K (P@K) BLI measure; see also §4.

[2] We linearly scale f_C scores to the range of $[0, 1]$. In the rest of this paper, unless stated otherwise, we assume that all f_C scores are already scaled.

are based on the most frequent N_{freq} source and target words in their respective vocabularies, where we conduct both forward and backward BLI for each of the N_{freq} most frequent words in $\mathcal{X}$ and $\mathcal{Y}$, and then retain word pairs with the highest CSLS matching scores. The final augmentation set $\mathcal{D}_{aug}$ is obtained after removing the duplicates and word pairs that contradict the pairs provided by $\mathcal{D}_S$. The final set of positives is then $\mathcal{D}_P = \mathcal{D}_S \cup \mathcal{D}_{aug}$.

In our preliminary analyses of CSLS similarity scores within the original CLWE space, we have detected that some non-translation pairs actually produce similar or even higher absolute CSLS scores than the corresponding ground truth positive pairs. Providing such information of *hard negative pairs*, collected into the set of negatives $\mathcal{D}_N$, would be a strong signal for CE fine-tuning: the core idea is that the cross-encoder will be able to 'overturn' such wrong predictions from the original CLWE space. In practice, for each $(w^x_+, w^y_+) \in D_P$, we propose to retrieve their respective negative words w^x_-, w^y_- that satisfy the following:

$$\begin{cases} f_C(w^x_+, w^y_-) \geqslant f_C(w^x_+, w^y_+) - \delta, \\ f_C(w^x_-, w^y_+) \geqslant f_C(w^x_+, w^y_+) - \delta, \end{cases} \quad (2)$$

where δ is a tunable margin. For a positive pair, we include at most N_{neg} w^x_- and w^y_- words to build negative pairs $(w^x_+, w^y_-), (w^x_-, w^y_+) \in \mathcal{D}_N$. We exclude pairs that already exist in $\mathcal{D}_P$, which occasionally occurs due to polysemy; that is, it holds $\mathcal{D}_P \cap \mathcal{D}_N = \emptyset$.

We also define a `reverse` operation $(\cdot)^\star$: $\pi^\star = (w^x, w^y)^\star = (w^y, w^x)$. Similarly, we extend the definition on sets such that the `reverse` of a set is a set of all its elements reversed, e.g., $\mathbf{\Pi}^\star = \mathcal{Y} \times \mathcal{X}$. It is evident that when using a set of fixed CLWEs, it holds $f_C(\pi) = f_C(\pi^\star)$ (Eq. 1). However, CEs are sensitive to the order of languages: for symmetry, we thus also provide $\mathcal{D}^\star_P$ and $\mathcal{D}^\star_N$ for CE fine-tuning. Due to the imbalance between $\mathcal{D}_P$ and $\mathcal{D}_N$, we repeat each positive pair (in both $\mathcal{D}_P$ and $\mathcal{D}^\star_P$) N_{rep} times for CE fine-tuning. The choice of the value N_{rep} impacts the distribution of word pairs for CE fine-tuning: $\pi \sim p_\pi$.

CE Fine-Tuning with Polarised Similarity Scores. As mentioned, we observed that in the original CLWE space there are frequent cases where a true translation pair *from the training set* obtains a lower CSLS score than one or more hard negative pairs (i.e., non-translations): $f_C(w^x_+, w^y_+) < f_C(w^x_+, w^y_-)$. Such cases can be fixed or mitigated

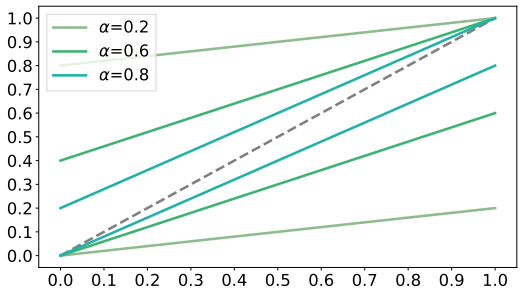

Figure 2: Linear polarisation functions with different values of the hyper-parameter α.

by *polarising* similarity scores for the word pairs in the sets $\mathcal{D}_P$ and $\mathcal{D}_N$ (and also $\mathcal{D}^\star_P$ and $\mathcal{D}^\star_N$) before CE fine-tuning. The polarisation step in practice means **1)** increasing the score $f_C(w^x_+, w^y_+)$ of positive pairs, and **2)** decreasing $f_C(w^x_+, w^y_-)$. The key rationale behind polarisation is exactly the following: cross-encoders can resolve or mitigate the difficult cases of highly similar hard negatives, where polarisation enables us to provide the correct learning signal for this purpose.

In practice, we construct a pair of monotonically increasing polarisation functions $g^+, g^- : \mathbb{R} \to \mathbb{R}$ to adjust the CSLS scores from the original CLWE space, such that for a positive word pair it holds $g^+ \circ f_C(\pi) \geqslant f_C(\pi)$; for a negative word pair it holds $g^- \circ f_C(\pi) \leqslant f_C(\pi)$. We adopt a pair of simple linear functions, where their domains and codomains are within the $[0, 1]$ interval, which is the same interval of the output values of CEs.[3] The functions are as follows:

$$\begin{cases} g^+(z) = \alpha \cdot z - \alpha + 1, \\ g^-(z) = \alpha \cdot z. \end{cases} \quad (3)$$

$\alpha \in [0, 1]$ is a hyper-parameter, with its impact on the polarisation function illustrated in Figure 2.When $\alpha = 1$, the original unaltered CSLS scores are used to fine-tune multilingual LMs; when $\alpha = 0$, binary labels (1: true translation pairs , 0: negative pairs) are used for CE fine-tuning. The linear polarisation function can be seen as a more expressive function that generalises over the two special cases.[4]

We then use the adjusted (i.e., polarised) scores to fine-tune multilingual pretrained LMs. For a

[3] This is because the sigmoid function is applied on the output logits of CEs.

[4] In our preliminary experiments, we also investigated non-linear polynomials as polarisation functions, but such functions led to only small to negligible BLI performance gains; we thus omit them for brevity.

word pair π, we denote the CE prediction as $u_\pi = \sigma(f_\theta(\pi))$, where $\sigma(\cdot)$ is the sigmoid function. The polarised similarity score for fine-tuning is denoted as $v_\pi = g^{+/-} \circ f_C(\pi)$, where $g^{+/-}(\pi)$ is $g^+(\pi)$ if $\pi \in \mathcal{D}_P \cup \mathcal{D}_P^\star$, and $g^-(\pi)$ if $\pi \in \mathcal{D}_N \cup \mathcal{D}_N^\star$. θ represents the entire set of parameters of the cross-encoder. We then train BLICEr with the standard Binary Cross-Entropy loss:

$$\mathcal{L} = -\mathbb{E}_{\pi \sim p_\pi}[v_\pi \cdot \log(\sigma(u_\pi)) + (1 - v_\pi) \cdot \log(1 - \sigma(u_\pi))].$$

Providing Text Input to Cross-Encoders. Cross-encoders must 'consume' word pairs π as input. We enclose the word pairs into text templates that are fed to the CE. Given a word pair (w^x, w^y), the final input is then in the format $T(w^x), T(w^y)$, where a template-based transformation is applied to each word from the pair. There is a spectrum of possible templates (e.g., see Appendix B for a list of 16 templates we designed), and in our main experiments we rely on the following one: $T(w) = [w]$ ($[L_w]$)!, where $[w]$ is replaced by the actual word, and $[L_w]$ is the word for the language to which w belongs (i.e., more precisely w can be found in the corresponding vocabulary of language L_w) in that actual language. For instance, the CE input for an English-French word pair *(apple, pomme)* is *apple (english)!, pomme (français)!*.

In practice, each mPLM's dedicated tokeniser then splits the template text input of two words into two sequences of WordPiece/subword tokens, inserts the special separation token between them, and appends and prepends other special tokens.[5]

Combining Similarity Scores. At BLI inference, we combine the similarity scores computed by the fine-tuned CE with the original CLWE scores, via a standard linear interpolation:

$$\widetilde{f}_\theta(\pi) = \frac{\sigma(f_\theta(\pi)) + \sigma(f_\theta(\pi^\star))}{2}. \quad (4)$$

$$f_{\text{Mix}}(\pi) = (1 - \lambda) f_C(\pi) + \lambda \widetilde{f}_\theta(\pi), \quad (5)$$

where $\lambda \in [0, 1]$ is a tunable hyper-parameter.

CE Reranking. One major drawback of cross-encoders is their high computation overhead for retrieval tasks with a large number of items in the target set (Karpukhin et al., 2020; Humeau et al., 2020; Geigle et al., 2022, *inter alia*). In particular, given a source word w^x, in order to predict its translation in $\mathcal{Y}$, CE models need to calculate similarity scores over all candidate pairs in $\mathcal{Y}$. We thus follow the body of work in information retrieval (Lin et al., 2021), and adopt a more efficient, two-stage '*retrieve-and-rerank*' approach (Karan et al., 2020; Geigle et al., 2022). First, we use more efficient CLWEs to retrieve the $N_{cand} \ll |\mathcal{Y}|$ candidate pairs with the highest similarity scores, and then rerank them relying on the additional knowledge from the CEs, based on Eq. 5. Feeding only the N_{cand} pairs to the CEs substantially decreases the computational overhead.

4 Experimental Setup

4.1 BLI Setups and Datasets

We use two standard and established BLI datasets: **1)** XLING (Glavaš et al., 2019) and **2)** PanLex-BLI (Vulić et al., 2019). XLING provides BLI training and test lexicons covering 8 languages from diverse language families (Croatian: HR, English: EN, Finnish: FI, French: FR, German: DE, Italian: IT, Russian: RU, Turkish: TR). We consider all 14 EN$\rightarrow*$ and $*\rightarrow$EN BLI directions. Further, for each BLI direction, we run experiments in *supervised* settings (i.e., where the training set $\mathcal{D}_S$ covers 5k word pairs) and *semi-supervised* settings (i.e., $|\mathcal{D}_S|$ = 1k). PanLex-BLI is a BLI benchmark oriented towards low-resource languages. We use a subset of PanLex-BLI comprising six diverse languages (Bulgarian: BG, Catalan: CA, Estonian: ET, Georgian: KA, Hebrew: HE, Hungarian: HU), yielding a total of 30 BLI directions. Since we deal with lower-resource languages in PanLex-BLI, we consider only semi-supervised setups with 1k pairs for training. Both XLING and PanLex-BLI trim the vocabularies of each language to the most frequent 200k words, and the standard choice of pretrained fastText word embeddings (Bojanowski et al., 2017) is used to derive CLWEs (see also Appendix F).

Both datasets provide a test set of 2k work pairs for each language pair, without any overlap with the training pairs. We report the standard *Precision@1* (P@1) scores.[6] CSLS with k=10 (see Eq. 1) is used as the CLWE-based similarity function.

[5] Different CEs may have different input text formats. XLM-R: `[s] ... [/s] [/s] ... [/s]`; mBERT: `[CLS] ... [SEP] ... [SEP]`. In the two examples ... denotes a subword sequence, `[s]` and `[CLS]` are special prefix tokens, and `[/s]` and `[SEP]` are special separation/suffix tokens.

[6] We observed very similar performance trends for P@5 and Mean Reciprocal Rank (MRR) as BLI measures.

4.2 CLWE Models (Baselines)

Our BLI method is evaluated against a representative set of strong SotA BLI models from recent literature; all of them are CLWE-based and with publicly available implementations. Here, we provide brief summaries:[7]

RCSLS (Joulin et al., 2018) is a representative CLWE model trained directly with the BLI-style objective and displays strong performance in supervised BLI tasks. It first learns an initial mapping via Procrustes (Xing et al., 2015) and then fine-tunes the mapping via a relaxed CSLS loss.

VecMap (Artetxe et al., 2018) leverages a self-learning procedure and demonstrates strong performance in unsupervised and semi-supervised BLI settings. It also supports unsupervised BLI. For the supervised setting, its self-learning is switched off, producing better BLI results.

ContrastiveBLI (Li et al., 2022) is the current SotA BLI model outperforming all existing methods in supervised and semi-supervised BLI settings. It is a two-stage model where both stages, termed **C1** and **C2**, leverage contrastive fine-tuning. Stage C1 is based purely on static CLWEs (i.e., CLWEs derived from static WEs such as fastText) and it refines an initial CLWE mapping via BLI-oriented contrastive fine-tuning with a self-learning procedure, attracting true translation pairs together and pushing away hard negative pairs. Stage C2 first derives 'decontextualised' mBERT word embeddings via contrastively tuning a pretrained mBERT model, and then linearly combines C1-induced CLWEs with mBERT-based word vectors. C2 typically further improves BLI performance over C1's output. In this work, we provide comparisons against CLWEs from both stages.[8]

For all the baselines, we follow their original suggested settings and hyper-parameter choices for supervised and semi-supervised BLI settings.

[7]For further technical details and descriptions of each BLI model, we refer to their respective publications. We used the publicly available implementations of all the baseline models.

[8]Li et al. (2022) empirically validated that other standard BLI methods such as LNMap (Mohiuddin et al., 2020) or FIPP (Sachidananda et al., 2021) yield BLI performances which are on average similar to or weaker than those obtained by RCSLS and VecMap. Moreover, all the methods: RCSLS, VecMap, FIPP, and LNMap, are consistently outperformed by ContrastiveBLI's C1 and C2 stages (Li et al., 2022), which serve as our strongest BLI baselines. Because of that and for clarity, we omit FIPP and LNMap from the experiments.

4.3 BLICEr: Training Setup and Hyper-parameters

Training Setup. Since BLI datasets typically do not provide separate development sets, previous work conducted hyper-parameter search on a randomly selected language pair (Glavaš et al., 2019; Karan et al., 2020; Li et al., 2022) from the BLI benchmark. We adopt this approach, and tune BLICEr's hyper-parameters on the (EN,TR) pair from XLING. For PanLex-BLI, we inherit all hyper-parameter values from the XLING experiments, and only further tune the λ value on the randomly sampled (HU,KA) pair. We also select the final text template (see §3 and Appendix B) in the same fashion.

Multilingual Pretrained Language Models. We test three mPLMs: mBERT (Devlin et al., 2019), XLM-R$_{base}$, and XLM-R$_{large}$ (Conneau et al., 2020). Unless noted otherwise, XLM-R$_{large}$ is used as the main model for BLICEr.

Hyper-parameters. For supervised setups, BLICEr is fine-tuned for 3 epochs, N_{aug}=0, N_{rep}=8, δ=0.1, and α=0.7; for semi-supervised setups, BLICEr is trained for 5 epochs, N_{aug}=4k, N_{rep}=4, δ=0.2, and α=1.0. The λ values for different setups are listed in Appendix E. In all BLI setups, we use AdamW (Loshchilov and Hutter, 2019) with the learning rate of $1.2e$-5, and the weight decay is 0.01; the maximum sequence length is 20, the batch size is 256, N_{freq}=20k, N_{neg}=28, and N_{cand}=28 (see §3 again for the description of each hyper-parameter).

5 Results and Discussion

The main results on XLING and PanLex-BLI are presented in Table 1 and Table 2, with additional results available in Appendix D). The tables span $14 + 30 = 44$ BLI directions, in supervised and semi-supervised scenarios, and with four CLWE methods respectively, which yields a total of 352 different BLI setups. One major quantitative finding is that the proposed BLICEr method derives gains in $351/352$ setups. In what follows, we delve deeper into the analyses across multiple aspects.

Supervised and Semi-Supervised Setups. Our results on XLING demonstrate that BLICEr yields outstanding performance in both supervision setups. In the 5k-setup, the average gain over four CLWE models is 7.76 P@1 points, and the value is 7.35 for the 1k-setup. Combining BLICEr with the two strongest baseline BLI models, C1 and

[5k] **Pairs**	**EN→DE**	**DE→EN**	**EN→FI**	**FI→EN**	**EN→FR**	**FR→EN**	**EN→HR**	**HR→EN**	**EN→IT**	**IT→EN**	**EN→RU**	**RU→EN**	**EN→TR**	**TR→EN**	**Avg.**
RCSLS	57.60	56.55	42.05	41.25	66.55	63.11	37.90	35.67	64.05	61.50	49.40	48.66	39.05	37.43	50.06
RCSLS + BLICEr	**64.00**	58.95	53.60	52.60	**71.75**	66.17	53.15	48.92	**70.50**	65.79	**60.45**	56.26	50.35	45.74	58.44
VECMAP	51.00	55.24	37.75	43.51	63.10	62.75	34.05	39.08	60.40	62.17	39.65	49.35	32.05	39.24	47.81
VECMAP + BLICEr	59.95	58.16	53.05	53.65	69.70	65.44	54.60	52.55	69.80	65.79	56.95	55.53	48.65	46.17	57.86
C1	54.90	57.64	44.50	46.24	65.05	63.84	40.60	42.29	63.45	63.57	49.15	51.86	41.35	42.60	51.93
C1 + BLICEr	62.75	59.68	54.25	54.02	70.75	66.48	55.40	53.55	70.05	66.10	59.25	57.41	51.05	48.14	59.21
C2	58.05	59.31	47.15	49.97	67.55	65.39	47.85	49.13	65.25	64.65	50.80	55.21	45.05	44.46	54.99
C2 + BLICEr	63.45	**60.67**	**55.95**	**55.33**	70.90	**67.36**	**57.55**	**55.65**	70.25	**66.87**	60.40	**58.25**	**52.85**	**48.88**	**60.31**
[1k] **Pairs**	**EN→DE**	**DE→EN**	**EN→FI**	**FI→EN**	**EN→FR**	**FR→EN**	**EN→HR**	**HR→EN**	**EN→IT**	**IT→EN**	**EN→RU**	**RU→EN**	**EN→TR**	**TR→EN**	**Avg.**
RCSLS	46.10	48.25	28.35	28.38	56.50	55.56	22.50	22.88	55.20	53.64	35.50	36.62	23.00	24.65	38.37
RCSLS + BLICEr	**56.50**	55.97	45.90	44.56	63.65	61.87	41.10	40.03	64.45	60.83	52.25	49.40	40.20	38.55	51.09
VECMAP	48.25	54.25	27.75	41.30	60.30	61.25	25.50	37.56	57.45	60.88	24.80	46.31	26.55	37.11	43.52
VECMAP + BLICEr	50.50	57.43	33.30	51.92	63.35	65.29	37.75	51.76	61.00	64.50	28.60	52.80	34.40	46.22	49.92
C1	50.45	56.29	42.15	45.35	61.65	63.27	35.65	40.77	59.50	62.74	42.55	50.34	38.10	42.23	49.36
C1 + BLICEr	52.50	**59.36**	**50.95**	**54.02**	**64.40**	**65.75**	49.30	53.34	**65.05**	**65.12**	50.80	56.21	**46.55**	**48.40**	**55.84**
C2	51.00	57.17	44.45	48.34	62.05	64.25	42.35	46.82	61.35	64.03	46.15	53.17	41.30	43.56	51.86
C2 + BLICEr	51.05	58.95	50.15	53.91	63.00	65.24	**50.90**	**54.81**	62.85	64.65	**52.70**	**56.68**	46.35	47.82	55.65

Table 1: BLI scores (P@1×100%) on the XLING BLI benchmark in supervised and semi-supervised scenarios. We apply the proposed BLICEr method to all four CLWE-based baselines (i.e., the 'baseline + BLICEr' rows). The scores in **bold** denote the highest score per column and supervision setup.

[1k] **Pairs**	**BG→∗**	**∗→BG**	**CA→∗**	**∗→CA**	**HE→∗**	**∗→HE**	**ET→∗**	**∗→ET**	**HU→∗**	**∗→HU**	**KA→∗**	**∗→KA**	**Avg.**
RCSLS	14.97	15.36	12.61	13.54	9.37	7.57	10.30	10.58	14.48	14.30	6.80	7.18	11.42
RCSLS + BLICEr	30.92	31.15	26.31	26.84	19.73	17.43	23.96	25.64	28.96	28.51	17.69	18.01	24.60
VECMAP	32.25	31.35	26.08	32.62	26.06	24.41	26.43	23.94	30.65	33.56	22.76	18.35	27.37
VECMAP + BLICEr	42.15	40.60	32.39	39.20	34.14	35.54	38.05	34.75	38.38	41.20	33.90	27.70	36.50
C1	35.96	34.97	31.41	35.04	26.37	25.69	28.06	26.85	36.45	36.38	22.34	21.65	30.10
C1 + BLICEr	47.21	44.75	41.31	43.21	36.31	37.67	41.21	40.45	45.28	44.21	34.73	35.77	41.01
C2	40.14	38.98	35.67	39.41	30.05	29.51	33.31	33.21	39.78	38.89	25.22	24.17	34.03
C2 + BLICEr	**48.29**	**45.88**	**42.76**	**44.46**	**38.25**	**39.07**	**43.23**	**42.82**	**45.74**	**44.86**	**36.06**	**37.23**	**42.39**

Table 2: BLI scores (P@1×100%) on PanLex-BLI. $L \rightarrow *$ and $* \rightarrow L$ denote the average scores where L is the source and target language respectively. Detailed results for each language pair are in Appendix D.

[5k] **Pairs**	**EN→∗**	**∗→EN**	**Avg.**
C1	51.29	52.58	51.93
C1 + BLICEr (off-the-shelf)	50.9	52.40	51.65
C1 + BLICEr (w/o Template)	59.81	57.51	58.66
C1 + BLICEr	**60.50**	**57.91**	**59.21**
C2	54.53	55.45	54.99
C2 + BLICEr (off-the-shelf)	54.46	55.51	54.99
C2 + BLICEr (w/o Template)	61.31	58.39	59.85
C2 + BLICEr	**61.62**	**59.00**	**60.31**
[1k] **Pairs**	**EN→∗**	**∗→EN**	**Avg.**
C1	47.15	51.57	49.36
C1 + BLICEr (off-the-shelf)	47.14	51.59	49.37
C1 + BLICEr (w/o Template)	53.96	56.89	55.43
C1 + BLICEr	**54.22**	**57.46**	**55.84**
C2	49.81	53.91	51.86
C2 + BLICEr (off-the-shelf)	49.81	53.91	51.86
C2 + BLICEr (w/o Template)	**53.94**	57.00	55.47
C2 + BLICEr	53.86	**57.44**	**55.65**

Table 3: Ablation study. P@1×100% scores.

C2, yields average gains of 6.3 (5k-setup) and 5.14 points (1k-setup). The baseline BLI scores in the 5k-setup are already much higher than in the 1k-setup, intuitively offering less room for further performance boosts. However, we observe substantial gains with BLICEr in the 5k-setup across the board, suggesting that BLICEr effectively leverages the more abundant 'gold' bilingual supervision in the 5k-setup, as well as the 'silver' supervision derived from the CLWE space, which is more accurate in the 5k-setup than in the 1k-setup.

Compatibility with Different CLWEs. The results indicate that BLICEr is compatible with all CLWE baselines. The 'C2 + BLICEr' model achieves the highest average score in the XLING (5k) and PanLex-BLI (1k) setups. The 'C1 + BLICEr' variant is the best-performing one in the XLING (1k) setup. Overall, we observe a general trend: **1)** BLICEr derives a larger absolute gain when applied to a weaker input CLWE space, but **2)** starting from a stronger CLWE backbone still yields a stronger 'CLWE + BLICEr' model in terms of the absolute BLI performance.[9]

Performance over Languages. The results further indicate that the usefulness of BLICEr, while observed for all language pairs, is especially pro-

[9]There are still some slight deviations from the general trend: e.g., the baseline VECMAP outperforms RCSLS in the XLING (1k) setup on average, but 'RCSLS + BLICEr' surpasses 'VECMAP + BLICEr' by 1.17 points.

nounced for typologically more distant and lower-resource language pairs. The average gain over four CLWE backbones in the PanLex-BLI (1k) experiments is 10.4 P@1 points. Directly modeling interaction between two text items (e.g., two words turned into two text templates in our case) which allows them to learn finer-grained 'interaction features' (Thakur et al., 2021; Geigle et al., 2022), CEs seem especially important in low-resource setups.

Ablation Study and Further Analysis. We now study the effectiveness of each key component of BLICEr, basing our analyses on the best-performing baseline CLWEs: C1 and C2.

First, *the effectiveness of CE fine-tuning* is indicated by the results in Table 3. Combining off-the-shelf mPLMs with the baseline CLWEs derives no gains at all. We further investigate *the usefulness of templates* (see §3), with results also provided in Table 3. In general, the scores indicate that templates are not crucial for BLI performance, and simply providing two words to the CE without any extra template also yields very strong BLI performance across the board. We observe only slight average gains in both supervision setups. We also provide average results per each tested template in Appendix B, further suggesting that strong gains are achieved irrespective of the chosen template.

Further, we study *the impact of the underlying mPLM* on the final BLI performance, with the results summarised in Table 4. The scores render our BLICEr method useful with all mPLMs. As expected, the largest XLM-R$_{large}$ model yields the best performance, and we also note a slight edge of XLM-R$_{base}$ over mBERT.

Figure 3 further plots *the impact of polarisation*. First, we note that there are substantial gains with all the α values (cf. Table 1). In the 5k-setup, polarisation achieves a further average gain of 2 points (α=0.7 versus α=1.0). In the 1k-setup, it seems that using the original CLWE similarity scores without polarisation (i.e., α=1.0) yields slightly better results. We attribute this behaviour to potentially noisy 'silver' positive pairs in the 1k-setup, which might dilute the gold knowledge from $\mathcal{D}_S$ as the polarisation step might amplify the noise further. Five times more abundant 'gold' supervision and more reliable CLWEs in the 5k-setup yield a stronger learning signal for BLICEr, and this undesirable phenomenon then gets mitigated.

Finally, Figure 4 demonstrates *the impact of interpolation* of the CE-based scores and the original

[5k] Pairs	EN→∗	∗→EN	Avg.
C1	51.29	52.58	51.93
C1 + BLICEr (mBERT)	55.25	54.84	55.05
C1 + BLICEr (XLM-R$_{base}$)	57.09	55.76	56.43
C1 + BLICEr (XLM-R$_{large}$)	**60.50**	**57.91**	**59.21**
C2	54.53	55.45	54.99
C2 + BLICEr (mBERT)	56.29	55.82	56.06
C2 + BLICEr (XLM-R$_{base}$)	57.59	56.57	57.08
C2 + BLICEr (XLM-R$_{large}$)	**61.62**	**59.00**	**60.31**
[1k] Pairs	**EN→∗**	**∗→EN**	**Avg.**
C1	47.15	51.57	49.36
C1 + BLICEr (mBERT)	50.36	54.25	52.30
C1 + BLICEr (XLM-R$_{base}$)	51.46	55.17	53.31
C1 + BLICEr (XLM-R$_{large}$)	**54.22**	**57.46**	**55.84**
C2	49.81	53.91	51.86
C2 + BLICEr (mBERT)	50.28	54.58	52.43
C2 + BLICEr (XLM-R$_{base}$)	51.19	55.38	53.28
C2 + BLICEr (XLM-R$_{large}$)	**53.86**	**57.44**	**55.65**

Table 4: BLICEr based on different pretrained LMs.

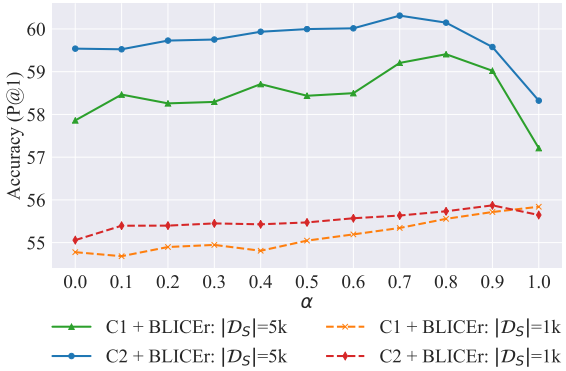

Figure 3: Average BLI scores on XLING with different values of the polarisation hyper-parameter α.

CLWE scores. The bell-shaped curves across different BLI setups **1)** clearly indicate the synergistic effect and the usefulness of interpolation across the board, and also **2)** show that the (near-)optimal λ values depend on the amount of supervision. In the 5k-setup, the peak λ values put more weight to the original CLWE space (λ=0): this is expected as the starting CLWE space is more accurate when induced with more 'gold' supervision, and the CE-based knowledge helps to a lesser extent. The peak λ values are higher for the resource-leaner 1k-setup. Figure 4 also reveals that using only the CE output (λ=1) yields sub-optimal BLI performance, and the true benefit of CE fine-tuning is displayed only in the synergy with the original CLWE space.

Unsupervised and Zero-Shot Setups. While this paper mainly focuses on arguably more practical supervised and semi-supervised settings,[10] we also

[10]For instance, Vulić et al. (2019) empirically prove that using even a small amount of supervision (e.g., 200, 500 or 1,000 word translation pairs) always outperforms fully

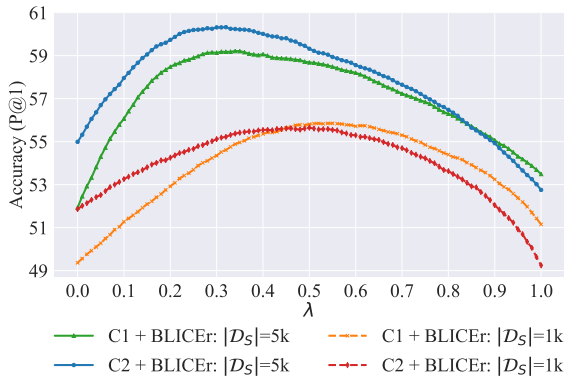

Figure 4: Average BLI scores on XLING with different values of the interpolation hyper-parameter λ.

conduct preliminary investigations of BLICEr in *fully unsupervised* and *zero-shot* settings, where *no direct bilingual supervision* between the source and the target is assumed: see Appendix C for the overview of the experimental setup. The results in these extreme settings, provided in Appendix C, further validate the usefulness of BLICEr as a post-processing method: it again yields substantial and consistent gains when applied to the backbone fully unsupervised VECMAP CLWE space.

6 Conclusion

We presented BLICEr, a simple and effective *post-hoc* reranking method for improved bilingual lexicon induction (BLI). BLICEr is applicable to any underlying cross-lingual word embedding (CLWE) space. It is based on fine-tuning multilingual pretrained language models into BLI-oriented cross-encoders with a limited amount of direct bilingual supervision (i.e., seed word translation pairs). At BLI inference, the BLICEr output refines cross-lingual word similarities from the underlying CLWE space. We conducted extensive empirical studies covering a total of 352 supervised and semi-supervised BLI setups, and observe substantial gains against representative and strong BLI baselines across the board. We also performed a series of ablation studies and validated the unsupervised and zero-shot capabilities of BLICEr. In future research we plan to experiment with other multilingual language models (He et al., 2021) and their ensembles, and we will extend the work to other languages and multilingual lexical tasks.

unsupervised BLI methods, while Artetxe et al. (2020) and Wang et al. (2022) discuss that at least some word translation pairs such as PanLex dictionaries (Kamholz et al., 2014) are available for thousands of the world's languages.

Acknowledgements

This work has been partially supported by the ERC PoC Grant MultiConvAI (no. 957356) and a research donation from Huawei. YL and FL are supported by Grace & Thomas C. H. Chan Cambridge International Scholarship. IV is also supported by a personal Royal Society University Research Fellowship.

Limitations

There are almost 7,000 languages worldwide (Lewis, 2009). However, publicly available fastText word embeddings currently only cover 294 languages,[11] mBERT supports only 104 languages,[12] and XLM-R only 100.[13] More effort is needed towards building bilingual dictionaries and language technology tools for under-represented and low-resource languages. Researchers, in the future, may also consider to develop techniques to address low-resource languages even without enough monolingual data for pretraining language models. Our work does not extend the scope to additional languages, and is by proxy also constrained by the current limitations of the underlying models such as fastText, mBERT, and XLM-R.

Some of the existing established BLI datasets were built with publicly available translation tools such as Google Translate plus some simple *post-hoc* refinements (Lample et al., 2018; Glavaš et al., 2019). There are occasionally noisy data points in the supposedly 'gold standard' datasets (Kementchedjhieva et al., 2019), they are typically not fully adapted to languages with more productive morphosyntactic systems (Czarnowska et al., 2019), and the control of synonyms and polysemy is difficult. While these evaluation data deficiencies do not impact relative comparisons between BLI models, for real-world applications gold standard ground-truth data of higher quality are needed for a vast number of language pairs. Their careful creation and annotation should involve native speakers of (low-resource) target languages, bilingual speakers and linguists.

[11] https://fasttext.cc/docs/en/pretrained-vectors.html
[12] https://github.com/google-research/bert
[13] https://github.com/facebookresearch/XLM

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre. 2020. A call for more rigor in unsupervised cross-lingual learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*, pages 7375–7388, Online. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Jiecao Chen, Liu Yang, Karthik Raman, Michael Bendersky, Jung-Jung Yeh, Yun Zhou, Marc Najork, Danyang Cai, and Ehsan Emadzadeh. 2020. DiPair: Fast and accurate distillation for trillion-scale text matching and pair modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2925–2937, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*, pages 8440–8451, Online. Association for Computational Linguistics.

Paula Czarnowska, Sebastian Ruder, Edouard Grave, Ryan Cotterell, and Ann Copestake. 2019. Don't forget the long tail! a comprehensive analysis of morphological generalization in bilingual lexicon induction. 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'19)*, pages 974–983, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'19)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiangyu Duan, Baijun Ji, Hao Jia, Min Tan, Min Zhang, Boxing Chen, Weihua Luo, and Yue Zhang. 2020. Bilingual dictionary based neural machine translation without using parallel sentences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*, pages 1570–1579, Online. Association for Computational Linguistics.

Eric Gaussier, J.M. Renders, I. Matveeva, C. Goutte, and H. Dejean. 2004. A geometric view on bilingual lexicon extraction from comparable corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*, pages 526–533, Barcelona, Spain.

Gregor Geigle, Jonas Pfeiffer, Nils Reimers, Ivan Vulić, and Iryna Gurevych. 2022. Retrieve fast, rerank smart: Cooperative and joint approaches for improved cross-modal retrieval. *Transactions of the Association for Computational Linguistics*, 10:503–521.

Goran Glavaš, Robert Litschko, Sebastian Ruder, and Ivan Vulić. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL'19)*, pages 710–721, Florence, Italy. Association for Computational Linguistics.

Goran Glavaš and Ivan Vulić. 2020. Non-linear instance-based cross-lingual mapping for non-isomorphic embedding spaces. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*, pages 7548–7555, Online. Association for Computational Linguistics.

Hila Gonen, Shauli Ravfogel, Yanai Elazar, and Yoav Goldberg. 2020. It's not Greek to mBERT: Inducing word-level translations from multilingual BERT. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 45–56, Online. Association for Computational Linguistics.

Edouard Grave, Armand Joulin, and Quentin Berthet. 2019. Unsupervised alignment of embeddings with wasserstein procrustes. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics (AISTATS'19)*, volume 89 of *Proceedings of Machine Learning Research*, pages 1880–1890. PMLR.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. *CoRR*, abs/2111.09543.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020. ConveRT: Efficient and accurate conversational representations from transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2161–2174, Online. Association for Computational Linguistics.

Geert Heyman, Ivan Vulić, and Marie-Francine Moens. 2017. Bilingual lexicon induction by learning to combine word-level and character-level representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL'17)*, pages 1085–1095, Valencia, Spain. Association for Computational Linguistics.

Geert Heyman, Ivan Vulić, and Marie-Francine Moens. 2018. A deep learning approach to bilingual lexicon induction in the biomedical domain. *BMC Bioinformatics*, 19(1):259:1–259:15.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations (ICLR'20)*.

Pratik Jawanpuria, Arjun Balgovind, Anoop Kunchukuttan, and Bamdev Mishra. 2019. Learning multilingual word embeddings in latent metric space: A geometric approach. *Transactions of the Association for Computational Linguistics*, 7:107–120.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*, pages 2979–2984, Brussels, Belgium. Association for Computational Linguistics.

David Kamholz, Jonathan Pool, and Susan Colowick. 2014. PanLex: Building a resource for panlingual lexical translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3145–3150, Reykjavik, Iceland. European Language Resources Association (ELRA).

Mladen Karan, Ivan Vulić, Anna Korhonen, and Goran Glavaš. 2020. Classification-based self-learning for weakly supervised bilingual lexicon induction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*, pages 6915–6922, Online. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*, pages 6769–6781, Online. Association for Computational Linguistics.

Yova Kementchedjhieva, Mareike Hartmann, and Anders Søgaard. 2019. Lost in evaluation: Misleading benchmarks for bilingual dictionary induction. 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'19)*, pages 3336–3341, Hong Kong, China. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proceedings of the International Conference on Learning Representations (ICLR'18)*.

M. Paul Lewis, editor. 2009. *Ethnologue: Languages of the World*, sixteenth edition. SIL International.

Yaoyiran Li, Fangyu Liu, Nigel Collier, Anna Korhonen, and Ivan Vulić. 2022. Improving word translation via two-stage contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL'22)*, pages 4353–4374, Dublin, Ireland. Association for Computational Linguistics.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. *Pretrained Transformers for Text Ranking: BERT and Beyond*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Fangyu Liu, Yunlong Jiao, Jordan Massiah, Emine Yilmaz, and Serhii Havrylov. 2022. Trans-encoder: Unsupervised sentence-pair modelling through self-and mutual-distillations. In *International Conference on Learning Representations (ICLR'22)*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR'19)*.

Tomás Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Tasnim Mohiuddin, M Saiful Bari, and Shafiq Joty. 2020. LNMap: Departures from isomorphic assumption in bilingual lexicon induction through non-linear mapping in latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*, pages 2712–2723, Online. Association for Computational Linguistics.

Barun Patra, Joel Ruben Antony Moniz, Sarthak Garg, Matthew R. Gormley, and Graham Neubig. 2019. Bilingual lexicon induction with semi-supervision in non-isometric embedding spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL'19)*, pages 184–193, Florence, Italy. Association for Computational Linguistics.

Xutan Peng, Chenghua Lin, and Mark Stevenson. 2021. Cross-lingual word embedding refinement by ℓ_1 norm optimisation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'21)*, pages 2690–2701, Online. Association for Computational Linguistics.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'18)*, pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'21)*, pages 5835–5847, Online. Association for Computational Linguistics.

Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL'95)*, pages 320–322, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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'19)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Inteligence Research*, 65:569–631.

Vin Sachidananda, Ziyi Yang, and Chenguang Zhu. 2021. Filtered inner product projection for crosslingual embedding alignment. In *Proceedings of the International Conference on Learning Representations (ICLR'21)*.

Haoyue Shi, Luke Zettlemoyer, and Sida I. Wang. 2021. Bilingual lexicon induction via unsupervised bitext construction and word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP'21)*, pages 813–826, Online. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'21)*, pages 296–310, Online. Association for Computational Linguistics.

Jack Urbanek, Angela Fan, Siddharth Karamcheti, Saachi Jain, Samuel Humeau, Emily Dinan, Tim Rocktäschel, Douwe Kiela, Arthur Szlam, and Jason Weston. 2019. Learning to speak and act in a fantasy text adventure game. 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'19)*, pages 673–683, Hong Kong, China. Association for Computational Linguistics.

Ivan Vulić, Simon Baker, Edoardo Maria Ponti, Ulla Petti, Ira Leviant, Kelly Wing, Olga Majewska, Eden Bar, Matt Malone, Thierry Poibeau, Roi Reichart, and Anna Korhonen. 2020a. Multi-SimLex: A large-scale evaluation of multilingual and crosslingual lexical semantic similarity. *Computational Linguistics*, 46(4):847–897.

Ivan Vulić, Goran Glavaš, Roi Reichart, and Anna Korhonen. 2019. Do we really need fully unsupervised cross-lingual embeddings? 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'19)*, pages 4407–4418, Hong Kong, China. Association for Computational Linguistics.

Ivan Vulić, Goran Glavaš, Fangyu Liu, Nigel Collier, Edoardo Maria Ponti, and Anna Korhonen. 2022. Exposing cross-lingual lexical knowledge from multilingual sentence encoders. *CoRR*, abs/2205.00267.

Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. 2021. LexFit: Lexical fine-tuning of pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP'21)*, pages 5269–5283, Online. Association for Computational Linguistics.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020b. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*, pages 7222–7240, Online. Association for Computational Linguistics.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022. Expanding pretrained models to thousands more languages via lexicon-based adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL'22)*, pages 863–877, Dublin, Ireland. Association for Computational Linguistics.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'15)*, pages

1006–1011, Denver, Colorado. Association for Computational Linguistics.

Michelle Yuan, Mozhi Zhang, Benjamin Van Durme, Leah Findlater, and Jordan Boyd-Graber. 2020. Interactive refinement of cross-lingual word embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*, pages 5984–5996, Online. Association for Computational Linguistics.

Jinpeng Zhang, Baijun Ji, Nini Xiao, Xiangyu Duan, Min Zhang, Yangbin Shi, and Weihua Luo. 2021. Combining static word embeddings and contextual representations for bilingual lexicon induction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2943–2955, Online. Association for Computational Linguistics.

A Languages in Experiments

Family	Language	$[L_w]$	Code
Afro-Asiatic	Hebrew	עברית	HE
Germanic	English	english	EN
	German	deutsch	DE
Kartvelian	Georgian	ქართული	KA
Romance	Catalan	català	CA
	French	français	FR
	Italian	italiano	IT
Slavic	Bulgarian	български	BG
	Croatian	hrvatski	HR
	Russian	русский	RU
Turkic	Turkish	türkçe	TR
Uralic	Estonian	eesti	ET
	Finnish	suomi	FI
	Hungarian	magyar	HU

Table 5: Languages involved in our experiments, categorized by language family. We also show their ISO 639-1 codes and $[L_w]$ used in some text templates (see §3 and Appendix B).

B Text Templates

We experiment with 16 different templates as follows. Among the 16 templates, $T_1 - T_4$ are 4 basic templates. $T_5 - T_8$ add a quotation mark (i.e., ‘’) to each $[w]$. $T_9 - T_{12}$ add a full stop (i.e., ‘.’) at the end of the template, and $T_{13} - T_{16}$ append the exclamation mark (i.e., ‘!’) to each sequence.

T_1: $[w]$
T_2: the word $[w]$
T_3: $[w]$ ($[L_w]$)
T_4: the word $[w]$ in $[L_w]$
T_5: ‘$[w]$’
T_6: the word ‘$[w]$’
T_7: ‘$[w]$’ ($[L_w]$)
T_8: the word ‘$[w]$’ in $[L_w]$
T_9: $[w]$.
T_{10}: the word $[w]$.
T_{11}: $[w]$ ($[L_w]$).
T_{12}: the word $[w]$ in $[L_w]$.
T_{13}: $[w]$!
T_{14}: the word $[w]$!
T_{15}: $[w]$ ($[L_w]$)!
T_{16}: the word $[w]$ in $[L_w]$!

An analysis supplementing the main analysis in the main paper (see §5) shows the average results with each template; the model variant is ‘C2 + BLICEr’ evaluated on XLING (5k) covering 14 BLI directions. The results are summarized in Table 6.

First, among $T_1 - T_4$, there is a very minor variation in the results, where T_3 seems to be slightly better than the other three templates. Second, adding quotation marks ($T_5 - T_8$) results in decreased BLI performance when compared to the basic templates (or using no template at all). Third, adding a full stop or an exclamation mark ($T_9 - T_{16}$) does not have any real impact on the results. We again emphasize **1)** that in our main experiments we pick the template T_{15} that achieves best performance on a single language pair (EN,TR), which we also use to tune all the hyper-parameters, but **2)** not using any template (effectively using the template T_1) also yields very strong results across the board (see also the main paper).

T_1	T_2	T_3	T_4
59.85	59.59	60.15	60.04
T_5	T_6	T_7	T_8
58.60	58.38	58.34	58.47
T_9	T_{10}	T_{11}	T_{12}
59.16	59.80	60.12	59.88
T_{13}	T_{14}	T_{15}	T_{16}
59.85	59.82	**60.31**	59.97

Table 6: ‘C2 + BLICEr’ with different templates. P@1×100% scores (average over all 14 BLI directions in the XLING benchmark (5k pairs).

C Unsupervised and Zero-shot Setups

This paper mainly focuses on the more practical supervised and semi-supervised settings, but as mentioned in the main paper (see the last paragraph of §5) we also conduct preliminary investigations of BLICEr in *fully unsupervised* and *zero-shot* settings, where no direct bilingual supervision between the source and the target is assumed. We rely on the unsupervised variant of VECMAP, a strong unsupervised BLI method (Glavaš et al., 2019), as the CLWE backbone for BLICEr: among our four CLWE baselines, VECMAP is the only one that supports *fully unsupervised* BLI.

We consider BLI tasks between several (randomly sampled) language pairs from the XLING benchmark that do not include English as one of the languages, and dispose of any direct bilingual supervision. **First**, the *fully unsupervised* setup is in fact a variant of our semi-supervised setup: for CE fine-tuning we now use only ‘silver’ word translation pairs obtained from the unsupervised CLWE space. We rely on the same hyper-parameters as in the semi-supervised setups (see §4). **Second**, in the *zero-shot* setup, while translating from the

	DE→TR	TR→DE	FI→HR	HR→FI	IT→FR	FR→IT	Avg.
VECMAP	23.79	26.46	28.80	27.72	**65.22**	63.42	39.24
+BLICER (unsuper)	29.53	**35.78**	**37.78**	**37.19**	64.44	64.46	44.86
+BLICER (zero-shot)	**33.54**	35.68	35.16	37.09	63.10	**64.82**	**44.90**

Table 7: BLICER in unsupervised and zero-shot setups.

language L_x to L_y, we assume that we only possess sets of word translations for the language pairs (EN, L_x) and (EN, L_y). This experiment aims to verify if BLICER can effectively leverage the inherent multilinguality of the underlying mPLMs.[14] In particular, we assume 2k seed word pairs for (EN, L_x) and another 2k pairs for (EN, L_y).[15] For CE fine-tuning, we then use the $4k$ positive word pairs, together with negative pairs retrieved from the two CLWE spaces,[16] without any augmentation with 'silver' translation pairs. We adopt the hyperparameter values from the supervised setup (see §4 again).

The results in Table 7 demonstrate that BLICER is also effective in unsupervised and zero-shot settings, yielding substantial gains over the unsupervised VECMAP baseline on average. The only exception is the highly similar language pair IT-FR, where the baseline CLWE model already strikes extremely high P@1 performance. Importantly, the results in zero-shot setups validate that BLICER implicitly benefits from the multilingual information stored in the underlying XLM-R$_{large}$ model.

In addition, we also show 'VECMAP + BLICER' results on XLING EN→∗ and ∗→EN in *fully unsupervised* setups. Table 8 presents BLICER results where the unsupervised VECMAP model is used as the CLWE backbone. Note that the hyperparameters are also tuned on the language pair (EN, TR). Consequently, we point out that the 'VECMAP + BLICER' results below in Table 8 for the EN→TR and TR→EN are in fact not unsupervised (we include them only for completeness).

[14]The input domains during training and evaluation are totally different. The zero-shot capability of BLICER may show that the zero-shot training with (EN, L_x) and (EN, L_y) input can expose the word translation knowledge from mPLMs for BLI between the pair (L_x, L_y).

[15]We randomly sample the 2k pairs from the respective 5k training sets in XLING. We additionally ensure that there is no overlap of English words between the two sets. This constraint prevents naively deriving word pairs between L_x and L_y from the two seed dictionaries via using the same word as the pivot.

[16]The negatives are derived from the semi-supervised C2-based CLWE spaces which are based on the provided dictionaries of 2k pairs. We rely on semi-supervised C2 only for deriving bilingual supervision for CE fine-tune, where (EN, L_x) and (EN, L_y) input is fed to the CE. At BLI inference, we use instead cross-lingual word similarity scores obtained from unsupervised VECMAP for (L_x, L_y).

D BLI Results on PanLex-BLI for Individual Language Pairs

In Table 9, we present full BLI results per each PanLex-BLI language pair, while the results in the main paper are aggregated over a particular source or target language (see Table 2).

E Values of the λ Hyper-parameter

The hyper-parameter values of λ are tuned on (EN, TR) and (HU, KA) translation directions for XLING and PanLex-BLI, respectively; $\lambda \in \{0, 0.01, 0.02, ..., 0.98, 0.99, 1\}$. Here, we show the finally selected λ values in our experiments, spanning the results in §5 and Appendix C.

F Reproducibility Checklist

- **BLI Data**: BLI datasets used in our experiments are publicly available.[17] [18]
- **Static Word Embeddings**: We adopt the standard monolingual word embeddings for deriving CLWEs, used in a body of prior work on BLI. In fact, the XLING benchmark already provides a set of preprocessed fastText WEs trained on Wikipedia, which is then our starting point.[19] Panlex-BLI does not provide processed WEs, so we follow the original instructions from the authors and adopt fastText WEs trained on Common Crawl + Wikipedia.[20] The WEs are trimmed to the most frequent $200k$ words for each language. The same WEs are used for all CLWE baselines.
- **Pretrained LMs**: We derive CEs by fine-tuning pretrained LMs includings the mBERT variant 'bert-base-multilingual-uncased', and

[17]https://github.com/codogogo/xling-eval

[18]https://github.com/cambridgeltl/panlex-bli

[19]https://fasttext.cc/docs/en/pretrained-vectors.html

[20]https://fasttext.cc/docs/en/crawl-vectors.html

	EN→DE	DE→EN	EN→FI	FI→EN	EN→FR	FR→EN	EN→HR	HR→EN	EN→IT	IT→EN	EN→RU	RU→EN	EN→TR	TR→EN	Avg.
VECMAP	48.45	54.51	28.15	41.04	60.10	61.51	24.10	36.30	57.40	60.78	25.10	46.41	26.50	36.90	43.37
VECMAP + BLICER (unsuper)	**51.30**	**57.07**	**36.95**	**52.92**	**63.35**	**64.20**	**36.45**	**50.34**	**62.05**	**64.03**	**29.15**	**52.65**	**37.05**	**46.11**	**50.26**

Table 8: BLI scores (P@1×100%) on the XLING benchmark, EN→∗ and ∗→EN unsupervised BLI tasks. Unsupervised VECMAP is used as the CLWE backbone, and we use only 'silver' word translation pairs for cross-encoder fine-tuning (see Appendix C).

[1k] Pairs - First Half	BG→CA	BG→HE	BG→ET	BG→HU	BG→KA	CA→HE	CA→ET	CA→HU	CA→KA	HE→ET	HE→HU	HE→KA	ET→HU	ET→KA	HU→KA
RCSLS	18.40	10.86	14.92	19.44	11.25	9.36	10.39	18.12	6.66	5.51	10.77	3.47	15.57	6.55	7.99
RCSLS + BLICER	34.29	23.48	32.95	36.40	27.51	20.34	26.66	31.33	19.86	16.14	24.57	3.47	32.14	18.36	20.85
VECMAP	39.66	30.80	27.88	38.77	24.13	24.87	22.21	35.47	14.29	17.66	33.56	13.61	35.60	17.80	21.91
VECMAP + BLICER	41.88	44.96	41.13	46.64	36.13	31.67	32.88	39.67	19.58	28.68	39.75	24.41	44.53	27.87	30.52
C1	41.88	33.92	33.47	41.49	29.02	30.39	26.72	38.78	23.02	16.54	35.34	11.45	40.23	17.43	27.33
C1 + BLICER	45.39	46.08	49.19	49.54	45.86	40.62	41.29	44.31	36.17	30.82	41.41	25.07	47.99	32.08	**39.69**
C2	43.93	38.64	40.50	44.62	33.04	34.69	35.51	41.44	26.64	21.65	36.43	14.20	44.59	18.91	28.06
C2 + BLICER	**46.14**	**47.02**	**51.21**	**50.17**	**46.91**	**41.60**	**45.24**	**44.59**	**37.60**	**33.91**	**42.96**	**28.80**	**48.44**	**33.62**	39.24
[1k] Pairs - Second Half	**CA→BG**	**HE→BG**	**ET→BG**	**HU→BG**	**KA→BG**	**HE→CA**	**ET→CA**	**HU→CA**	**KA→CA**	**ET→HE**	**HU→HE**	**KA→HE**	**HU→ET**	**KA→ET**	**KA→HU**
RCSLS	18.53	14.22	15.01	18.95	10.10	12.87	9.34	19.91	7.17	5.03	9.73	2.87	15.81	6.27	7.59
RCSLS + BLICER	33.37	29.14	31.49	35.14	26.62	25.33	23.04	34.93	16.61	14.78	21.21	7.34	32.66	19.76	18.11
VECMAP	33.54	31.14	30.19	36.52	25.37	34.32	26.08	39.88	23.15	22.47	24.89	19.01	30.03	21.89	24.38
VECMAP + BLICER	38.19	39.31	43.73	46.31	35.47	38.55	37.40	45.71	32.48	36.71	30.83	**33.56**	38.51	32.54	35.42
C1	38.13	33.67	33.96	39.86	29.23	34.84	29.89	43.89	24.69	18.78	33.14	12.21	38.02	19.53	26.05
C1 + BLICER	44.19	42.19	46.26	**48.16**	42.96	**42.09**	43.43	49.45	35.72	36.28	41.74	23.62	47.37	33.55	37.78
C2	40.06	38.37	38.49	43.03	34.94	39.59	39.34	47.14	27.08	25.20	36.09	12.94	44.58	23.79	27.37
C2 + BLICER	**44.76**	**43.60**	**47.44**	47.98	**45.63**	41.97	**47.40**	**50.61**	**36.18**	**39.25**	**43.21**	24.28	**47.65**	**36.09**	**38.13**

Table 9: BLI scores (P@1×100%) on the PanLex-BLI benchmark, consiting of six lower-resource languages. We apply the proposed BLICER to the four CLWE baselines (compare the 'baseline + BLICER' results with the results from the 'raw' baselines).

λ Values	XLING: [5k] Pairs	XLING: [1k] Pairs	XLING: Unsupervised	XLING: Zero-shot	PanLex-BLI: [1k] Pairs
RCSLS + BLICER (XLM-R_{large})	0.29	0.82	-	-	0.74
VECMAP + BLICER (XLM-R_{large})	0.36	0.61	0.68	0.68	0.65
C1 + BLICER (mBERT)	0.18	0.38	-	-	-
C1 + BLICER (XLM-R_{base})	0.22	0.40	-	-	-
C1 + BLICER (XLM-R_{large})	0.35	0.51	-	-	0.65
C1 + BLICER (XLM-R_{large}, off-the-shelf)	0.82	0.66	-	-	-
C1 + BLICER (XLM-R_{large}, w/o Template)	0.22	0.46	-	-	-
C1 + BLICER (XLM-R_{large}, α=0.0)	0.09	0.20	-	-	-
C1 + BLICER (XLM-R_{large}, α=0.1)	0.15	0.14	-	-	-
C1 + BLICER (XLM-R_{large}, α=0.2)	0.11	0.21	-	-	-
C1 + BLICER (XLM-R_{large}, α=0.3)	0.20	0.24	-	-	-
C1 + BLICER (XLM-R_{large}, α=0.4)	0.21	0.32	-	-	-
C1 + BLICER (XLM-R_{large}, α=0.5)	0.32	0.28	-	-	-
C1 + BLICER (XLM-R_{large}, α=0.6)	0.19	0.29	-	-	-
C1 + BLICER (XLM-R_{large}, α=0.7)	0.35	0.30	-	-	-
C1 + BLICER (XLM-R_{large}, α=0.8)	0.37	0.33	-	-	-
C1 + BLICER (XLM-R_{large}, α=0.9)	0.45	0.43	-	-	-
C1 + BLICER (XLM-R_{large}, α=1.0)	0.43	0.51	-	-	-
C2 + BLICER (mBERT)	0.17	0.17	-	-	-
C2 + BLICER (XLM-R_{base})	0.15	0.21	-	-	-
C2 + BLICER (XLM-R_{large})	0.31	0.50	-	-	0.57
C2 + BLICER (XLM-R_{large}, off-the-shelf)	0.67	0	-	-	-
C2 + BLICER (XLM-R_{large}, w/o Template)	0.25	0.43	-	-	-
C2 + BLICER (XLM-R_{large}, α=0.0)	0.14	0.14	-	-	-
C2 + BLICER (XLM-R_{large}, α=0.1)	0.16	0.15	-	-	-
C2 + BLICER (XLM-R_{large}, α=0.2)	0.17	0.17	-	-	-
C2 + BLICER (XLM-R_{large}, α=0.3)	0.20	0.19	-	-	-
C2 + BLICER (XLM-R_{large}, α=0.4)	0.22	0.17	-	-	-
C2 + BLICER (XLM-R_{large}, α=0.5)	0.26	0.26	-	-	-
C2 + BLICER (XLM-R_{large}, α=0.6)	0.26	0.22	-	-	-
C2 + BLICER (XLM-R_{large}, α=0.7)	0.31	0.28	-	-	-
C2 + BLICER (XLM-R_{large}, α=0.8)	0.42	0.33	-	-	-
C2 + BLICER (XLM-R_{large}, α=0.9)	0.35	0.43	-	-	-
C2 + BLICER (XLM-R_{large}, α=1.0)	0.44	0.50	-	-	-

Table 10: λ values. The cells with '-' represent BLI setups not covered in our experiments: only unsupervised VECMAP is used for XLING unsupervised and zero-shot setups; we conduct ablation study and investigate model variants on XLING 5k and 1k setups only.

the XLM-R variants 'xlm-roberta-base' and 'xlm-roberta-large', all publicly available from the huggingface.co model hub.

- **Parameter Counts**: The number of parameters are $167,356,416$ for mBERT, $278,043,648$ for XLM-R_{base}, and $559,890,432$ for XLM-R_{large}.
- **Source Code**: We release our code at `https:`

//github.com/cambridgeltl/BLICEr.

- **Computing Infrastructure**: We run our code on Wilkes3, a cluster with 80 nodes. Each node has $4 \times$ Nvidia 80GB A100 GPUs and $128 \times$ CPU cores. All our experiments require only one node with $1\times$ GPU and $32\times$ CPU cores.
- **Software**: Slurm 20.11.8, Python 3.9.7, PyTorch 1.10.1, Transformers 4.15.0, and Sentence-Transformers 2.1.0.
- **Runtime (Wall Time)**: The BLICEr training (XLM-R$_{large}$, C2 as the CLWE backbone) on a language pair typically costs 10 minutes in both supervised and semi-supervised BLI setups. It takes 3 minutes for one BLI evaluation run.
- **Robustness**: We found that the improvements of BLICEr are robust over all language pairs with different random seeds, and thus use a fixed random seed 33 over all experiments.