Improving Word Translation via Two-Stage Contrastive Learning

Anonymous ACL submission

Abstract

Word translation or bilingual lexicon induction (BLI) is a key cross-lingual task, aiming to bridge the lexical gap between different languages. In this work, we propose a robust and effective two-stage contrastive learning framework for the BLI task. As Stage C1, we propose to refine standard cross-lingual linear maps between static word embeddings (WEs) via a contrastive learning objective; we also show how to integrate it into the self-learning procedure for even more refined cross-lingual maps. In Stage C2, we conduct BLI-oriented contrastive fine-tuning of mBERT, unlocking its word translation capability. We also show that static WEs induced from the 'C2-tuned' mBERT complement static WEs from Stage C1. Comprehensive experiments on standard BLI datasets for diverse languages and different experimental setups demonstrate substantial gains achieved by our framework. While the BLI method from Stage C1 already yields substantial gains over all state-of-the-art BLI methods in our comparison, even stronger improvements are met with the full two-stage framework: e.g., we report gains for 112/112 BLI setups, spanning 28 language pairs.

1 Introduction and Motivation

Bilingual lexicon induction (BLI) or word translation is one of the seminal and long-standing tasks in multilingual NLP (Rapp, 1995; Gaussier et al., 2004; Heyman et al., 2017; Shi et al., 2021, *inter alia*). Its main goal is learning translation correspondences across languages, with applications of BLI ranging from language learning and acquisition (Yuan et al., 2020; Akyurek and Andreas, 2021) to machine translation (Qi et al., 2018; Duan et al., 2020; Chronopoulou et al., 2021) and the development of language technology in low-resource languages and domains (Irvine and Callison-Burch, 2017; Heyman et al., 2018). A large body of recent BLI work has focused on the so-called *mapping-based* methods (Mikolov et al., 2013; Artetxe et al., 2018; Ruder et al., 2019).[1] Such methods are particularly suitable for low-resource languages and weakly supervised learning setups: they support BLI with only as much as few thousand word translation pairs (e.g., 1k or at most 5k) as the only bilingual supervision (Ruder et al., 2019).[2]

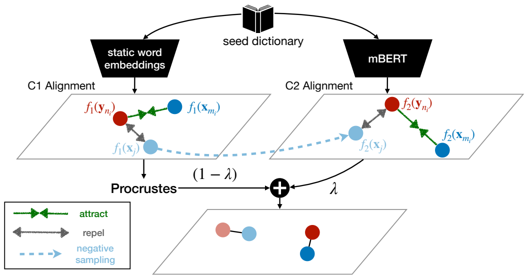

Figure 1: Illustration of the proposed two-stage BLI approach (see §2). It combines contrastive tuning on both static WEs (C1) and pretrained multilingual LMs (C2), where the static WEs are leveraged for selecting negative examples in contrastive tuning of the LM. Output of C1 and C2 is combined for the final BLI task.

Unlike for many other tasks in multilingual NLP (Doddapaneni et al., 2021; Chau and Smith, 2021; Ansell et al., 2021), state-of-the-art (SotA) BLI results are still achieved via static word embeddings (WEs) (Vulić et al., 2020b; Liu et al., 2021b). A typical *modus operandi* of mapping-based approaches is to first train monolingual WEs independently on monolingual corpora and then map them to a shared cross-lingual space via linear (Mikolov et al., 2013; Glavaš et al., 2019) or non-linear mapping func-

[1] They are also referred to as *projection-based* or *alignment-based* methods (Glavaš et al., 2019; Ruder et al., 2019).

[2] In the extreme, *fully unsupervised* mapping-based BLI methods can leverage monolingual data only without any bilingual supervision (Lample et al., 2018; Artetxe et al., 2018; Hoshen and Wolf, 2018; Mohiuddin and Joty, 2019; Ren et al., 2020, *inter alia*). However, comparative empirical analyses (Vulić et al., 2019) show that, with all other components equal, using seed sets of only 500-1,000 translation pairs, always outperforms fully unsupervised BLI methods. Therefore, in this work we focus on this more pragmatic (weakly) supervised BLI setup (Artetxe et al., 2020); we assume the existence of at least 1,000 seed translations per each language pair.

tions (Mohiuddin et al., 2020). In order to achieve even better results, many BLI methods also apply a self-learning loop where training dictionaries are iteratively (and gradually) refined, and improved mappings are then learned in each iteration (Artetxe et al., 2018; Karan et al., 2020). However, there is still ample room for improvement, especially for lower-resource languages and dissimilar language pairs (Vulić et al., 2019; Nasution et al., 2021).

On the other hand, another line of recent research has demonstrated that a wealth of lexical semantic information is encoded in large multilingual pretrained language models (LMs) such as mBERT (Devlin et al., 2019), but **1)** it is not straightforward to transform the LMs into multilingual lexical encoders (Liu et al., 2021b), **2)** extract word-level information from them (Vulić et al., 2020b, 2021), and **3)** word representations extracted from these LMs still cannot surpass static WEs in the BLI task (Vulić et al., 2020b; Zhang et al., 2021). Motivated by these insights, in this work we investigate following research questions:

(RQ1) Can we further improve (weakly supervised) mapping-based BLI methods based on static WEs?

(RQ2) How can we extract more useful cross-lingual word representations from pretrained multilingual LMs such as mBERT or XLM-R?

(RQ3) Is it possible to boost BLI by combining cross-lingual representations based on static WEs and the ones extracted from multilingual LMs?

Inspired by the wide success of contrastive learning techniques in *sentence-level* representation learning (Reimers and Gurevych, 2019; Carlsson et al., 2021; Gao et al., 2021), we propose a *two-stage contrastive learning framework for effective word translation* in (weakly) supervised setups; it leverages and combines multilingual knowledge from static WEs and pretrained multilingual LMs. **Stage C1** operates solely on static WEs: in short, it is a mapping-based approach with self-learning, where in each step we additionally fine-tune linear maps with contrastive learning that operates on gradually refined positive examples (i.e., true translation pairs), and hard negative samples. **Stage C2** fine-tunes a pretrained multilingual LM (e.g., mBERT), again with a contrastive learning objective, using positive examples as well as negative examples extracted from the output of C1. Finally, we extract word representations from the multilingual LM fine-tuned in Stage C2, and combine them with static cross-lingual WEs from Stage C1; the combined representations are then used for BLI.

We run a comprehensive set of BLI experiments on the standard BLI benchmark (Glavaš et al., 2019), comprising 8 diverse languages, in several setups. Our results indicate large gains over state-of-the-art BLI models: e.g., ≈+8 Precision@1 points on average, +10 points for many language pairs, gains for 107/112 BLI setups already after Stage C1 (cf., RQ1), and for all 112/112 BLI setups after Stage C2 (cf., RQ2 and RQ3). Moreover, our findings also extend to BLI for lower-resource languages from another BLI benchmark (Vulić et al., 2019). Finally, as hinted in recent work (Zhang et al., 2021), our findings validate that multilingual lexical knowledge in LMs, when exposed and extracted as in our contrastive learning framework, can complement the knowledge in static cross-lingual WEs (RQ3), and benefit BLI. We release the code and share the data at: `[URL]`.

2 Methodology

Preliminaries and Task Formulation. In BLI, we assume two vocabularies $\mathcal{X}=\{w^x_1,\ldots,w^x_{|\mathcal{X}|}\}$ and $\mathcal{Y}=\{w^y_1,\ldots,w^y_{|\mathcal{Y}|}\}$ associated with two respective languages L_x and L_y. We also assume that each vocabulary word is assigned its (static) type-level word embedding (WE), that is, the respective WE matrices for each vocabulary are $X \in \mathbb{R}^{|\mathcal{X}|\times d}$, $Y \in \mathbb{R}^{|\mathcal{Y}|\times d}$. d is WE dimensionality, with typical values $d = 300$ for static WEs (e.g., fastText) (Bojanowski et al., 2017), and $d = 768$ (mBERT and XLM-R WEs). We also assume a set of *seed* translation pairs $\mathcal{D}_0 = \{(w^x_{m_1}, w^y_{n_1}), ..., (w^x_{m_{|D_0|}}, w^y_{n_{|D_0|}})\}$ for training (Mikolov et al., 2013; Glavaš et al., 2019), where $1 \le m_i \le |\mathcal{X}|, 1 \le n_i \le |\mathcal{Y}|$. Typical values for the seed dictionary size $|\mathcal{D}_0|$ are $5k$ pairs and $1k$ pairs (Vulić et al., 2019), often referred to as *supervised* (5k) and semi-supervised or *weakly supervised* settings (1k) (Artetxe et al., 2018). Given another *test* lexicon $\mathcal{D}_T = \{(w^x_{t_1}, w^y_{g_1}), ..., (w^x_{t_{|\mathcal{D}_T|}}, w^y_{g_{|\mathcal{D}_T|}})\}$, where $\mathcal{D}_0 \cap \mathcal{D}_T = \emptyset$, for each L_x test word $w^x_{t_i}$ in $\mathcal{D}_T$ the goal is to retrieve its correct translation from the L_y's vocabulary $\mathcal{Y}$, and evaluate it against the gold L_y translation $w^y_{g_i}$ from the pair.

Method in a Nutshell. We propose a novel two-stage contrastive learning (CL) method, with both stages C1 and C2 realised via contrastive learning objectives, see Figure 1. Stage C1 (§2.1) operates solely on static WEs, and can be seen as a contrastive extension of mapping-based BLI approaches with static WEs. In practice, we blend contrastive learning with the standard SotA mapping-based framework with self-learning: VecMap (Artetxe et al., 2018), with some modifications. Stage C1 operates solely on static WEs in exactly the same BLI setup as prior work, and thus it can be evaluated independently. In Stage C2 (§2.2), we propose to leverage pretrained multilingual LMs for BLI, contrastively fine-tuning them for BLI, and extracting static WEs from the tuned LMs. These LM-based WEs can be combined with WEs obtained in Stage C1 (§2.3).

2.1 Stage C1

Stage C1 is based on the VecMap framework (Artetxe et al., 2018) which features **1)** *dual linear mapping*, where two separate linear transformation matrices map respective source and target WEs to a shared cross-lingual space; and **2)** a *self-learning* procedure that, in each iteration i refines the training dictionary and iteratively improves the mapping. We extend and refine VecMap's self-learning for supervised and semi-supervised settings via CL.

Initial Advanced Mapping. After ℓ_2-normalising word embeddings,[3] the two mapping matrices, denoted as $\boldsymbol{W}_x$ for the source language L_x and $\boldsymbol{W}_y$ for L_y, are computed via the Advanced Mapping (AM) procedure based on the training dictionary, as fully described in Appendix A.1; while VecMap leverages whitening, orthogonal mapping, re-weighting and de-whitening operations to derive mapped WEs, we compute $\boldsymbol{W}_x$ and $\boldsymbol{W}_y$ such that a one-off matrix multiplication produces the same result, see Appendix A.1 for the details.

Contrastive Fine-Tuning. At each iteration i, after the initial AM step, the two mapping matrices $\boldsymbol{W}_x$ and $\boldsymbol{W}_y$ are then further contrastively fine-tuned via the InfoNCE loss (Oord et al., 2018), a standard and robust choice of a loss function in CL research (Musgrave et al., 2020; Liu et al., 2021c,b). The core idea is to 'attract' aligned WEs of positive examples (i.e., true translation pairs) coming from the dictionary $\mathcal{D}_{i-1}$, and 'repel' *hard negative samples*, that is, words which are semantically similar but do not constitute a word translation pair.

These hard negative samples are extracted as follows. Let us suppose that $(w^x_{m_i}, w^y_{n_i})$ is a translation pair in the current dictionary $\mathcal{D}_{i-1}$, with its

[3]Unlike VecMap, we do not mean-center WEs as this yielded slightly better results in our preliminary experiments.

Algorithm 1 Stage C1: Self-Learning

Require: $\boldsymbol{X}$,$\boldsymbol{Y}$,$\mathcal{D}_0$,$\mathcal{D}_{\text{add}}$=$\emptyset$
for i=1:N_{iter} **do**
  $\boldsymbol{W}_x, \boldsymbol{W}_y \leftarrow$ Initial AM using D_{i-1};
  $\mathcal{D}_{\text{CL}} \leftarrow \mathcal{D}_0$ (supervised) or $\mathcal{D}_{i-1}$ (semi-super);
  for i=1:N_{CL} **do**
    Retrieve $\bar{\mathcal{D}}$ for the pairs from $\mathcal{D}_{\text{CL}}$;
    $\boldsymbol{W}_x, \boldsymbol{W}_y \leftarrow$ Optimize Contrastive Loss;
  Compute new $\mathcal{D}_{add}$;
  Update $\mathcal{D}_i = \mathcal{D}_0 \cup \mathcal{D}_{\text{add}}$;
return W_x, W_y

constituent words associated with d-dim static WEs $\mathbf{x}_{m_i}$ and $\mathbf{y}_{n_i}$. We then retrieve the nearest neighbours of $\mathbf{y}_{n_i}\boldsymbol{W}_y$ from $X\boldsymbol{W}_x$ and derive $\bar{w}^x_{m_i} \subset \mathcal{X}$ ($w^x_{m_i}$ excluded) , a set of hard negative samples of size N_{neg}. In a similar (symmetric) manner, we also derive the set of negatives $\bar{w}^y_{n_i} \subset \mathcal{Y}$ ($w^y_{n_i}$ excluded). We use $\bar{\mathcal{D}}_i$ to denote a collection of all hard negative set pairs over all training pairs in the current iteration i. We then fine-tune $\boldsymbol{W}_x$ and $\boldsymbol{W}_y$ by optimizing the following contrastive objective:

$$s_{i,j} = \exp(\cos(\mathbf{x}_i\boldsymbol{W}_x\ ,\ \mathbf{y}_j\boldsymbol{W}_y)/\tau), \tag{1}$$

$$p_i = \frac{s_{m_i,n_i}}{\sum\limits_{w^y_j \in \{w^y_{n_i}\} \cup \bar{w}^y_{n_i}} s_{m_i,j} + \sum\limits_{w^x_j \in \bar{w}^x_{m_i}} s_{j,n_i}}, \tag{2}$$

$$\min_{\boldsymbol{W}_x,\boldsymbol{W}_y} -\mathbb{E}_{(w^x_{m_i},w^y_{n_i})\in\mathcal{D}_{\text{CL}}} \log(p_i). \tag{3}$$

τ denotes a standard temperature parameter. The objective, formulated here for a single positive example, spans all positive examples from the current dictionary, along with the respective sets of negative examples computed as described above.

Self-Learning. The application of (a) initial mapping via AM and (b) contrastive fine-tuning can be repeated iteratively. Such self-learning loops typically yield more robust and better performing BLI methods (Artetxe et al., 2018; Vulić et al., 2019). At each iteration i, a set of automatically extracted high-confidence translation pairs $\mathcal{D}_{\text{add}}$ are added to the seed dictionary $\mathcal{D}_0$, and this dictionary $\mathcal{D}_i$ = $\mathcal{D}_0 \cup \mathcal{D}_{\text{add}}$ is then used in the next iteration $i+1$.

Our dictionary augmentation method slightly deviates from the one used by VecMap. We leverage the most frequent N_{freq} source and target vocabulary words, and conduct forward and backward dictionary induction (Artetxe et al., 2018). Unlike VecMap, we do not add stochasticity to the process, and simply select the top N_{aug} high-confidence word pairs from forward (i.e., source-to-target) induction and another N_{aug} pairs from the backward induction. In practice, we first retrieve the $2 \times N_{\text{aug}}$ pairs with the highest Cross-domain Similarity Local Scaling (CSLS) scores (Lample et al., 2018),[4] remove duplicate pairs and those that contradict with ground truth in $\mathcal{D}_0$, and add the rest into $\mathcal{D}_{\text{add}}$.

For the initial AM step, we always use the augmented dictionary $\mathcal{D}_0 \cup \mathcal{D}_{\text{add}}$; the same augmented dictionary is used for contrastive fine-tuning in weakly supervised setups.[5] We repeat the self-learning loop for N_{iter} times; in each iteration, we optimise the contrastive loss N_{CL} times, that is, we go N_{CL} times over all the positive pairs from the training dictionary (at this iteration). N_{iter} and N_{CL} are tunable hyper-parameters. Self-learning in Stage C1 is summarised in Algorithm 1.

2.2 Stage C2

Previous work tried to prompt off-the-shelf multilingual LMs for word translation knowledge via masked natural language templates (Gonen et al., 2020), averaging over their contextual encodings in a large corpus (Vulić et al., 2020b; Zhang et al., 2021), or extracting type-level WEs from the LMs directly without context (Vulić et al., 2020a, 2021). However, even sophisticated templates and WE extraction strategies still typically result in BLI performance inferior to fastText (Vulić et al., 2021).

(BLI-Oriented) Contrastive Fine-Tuning. Here, we propose to fine-tune off-the-shelf multilingual LMs relying on the supervised BLI signal: the aim is to expose type-level word translation knowledge directly from the LM, without any external corpora. In practice, we first prepare a dictionary of positive examples for contrastive fine-tuning: (a) $\mathcal{D}_{\text{CL}} = \mathcal{D}_0$ when $|\mathcal{D}_0|$ spans $5k$ pairs, or (b) when $|\mathcal{D}_0| = 1k$, we add the $N_{\text{aug}} = 4k$ automatically extracted highest-confidence pairs from Stage C1 (based on their CSLS scores, not present in $\mathcal{D}_0$) to $\mathcal{D}_0$ (i.e., $\mathcal{D}_{\text{CL}}$ spans $1k$ + $4k$ word pairs). We then extract N_{neg} hard negatives in the same way as in §2.1, relying on the shared cross-lingual space derived as the output of Stage C1. Our hypothesis is that a difficult task of discerning between true translation pairs and highly similar non-translations as hard negatives, formulated within a contrastive learning objective, will enable mBERT to expose its word translation knowledge, and complement the knowledge already available after Stage C1.

Throughout this work, we assume the use of pretrained mBERT$_{\text{base}}$ model with 12 Transformer layers and 768-dim embeddings.[6] Each raw word input w is tokenised, via mBERT's dedicated tokeniser, into the following sequence: $[CLS][sw_1] \dots [sw_M][SEP]$, $M \geq 1$, where $[sw_1] \dots [sw_M]$ refers to the sequence of M constituent subwords/WordPieces of w, and $[CLS]$ and $[SEP]$ are special tokens (Vulić et al., 2020b).

The sequence is then passed through mBERT as the encoder, its encoding function denoted as $f_\theta(\cdot)$: it extracts the representation of the [CLS] token in the last Transformer layer as the representation of the input word w. The full set of mBERT's parameters θ then gets contrastively fine-tuned in Stage C2, again relying on the InfoNCE CL loss:

$$s'_{i,j} = \exp(\cos(f_\theta(w_i^x), f_\theta(w_j^y))/\tau), \tag{4}$$

$$p'_i = \frac{s'_{m_i,n_i}}{\sum\limits_{w_j^y \in \{w_{n_i}^y\} \cup \bar{w}_{n_i}^y} s'_{m_i,j} + \sum\limits_{w_j^x \in \bar{w}_{m_i}^x} s'_{j,n_i}}, \tag{5}$$

$$\min_\theta - \mathbb{E}_{(w_{m_i}^x, w_{n_i}^y) \in \mathcal{D}_{\text{CL}}} \log(p'_i). \tag{6}$$

Type-level WE for each input word w is then obtained simply as $f_{\theta'}(w)$, where θ' refers to the parameters of the 'BLI-tuned' mBERT model.

2.3 Combining Output of C1 and C2

In order to combine the output WEs from Stage C1 and the mBERT-based WEs from Stage C2, we also need to map them into a 'shared' space: in other words, for each word w, its C1 WE and its C2 WE can be seen as two different views of the same data point. We thus learn an additional linear orthogonal mapping from the C1-induced cross-lingual WE space into the C2-induced cross-lingual WE space. It transforms ℓ_2-normed 300-dim C1-induced cross-lingual WEs into 768-dim cross-lingual WEs. Learning of the linear map $\boldsymbol{W} \in \mathbb{R}^{d_1 \times d_2}$, where in our case $d_1 = 300$ and $d_2 = 768$, is formulated as a Generalised Procrustes problem (Schönemann, 1966; Viklands, 2006) operating on all (i.e., both L_x and L_y) words from the seed translation dictionary $\mathcal{D}_0$.[7]

[4] Further details on the CSLS similarity and its relationship to cosine similarity are available in Appendix A.2.

[5] When starting with 5k pairs, we leverage only $\mathcal{D}_0$ for contrastive fine-tuning, as $\mathcal{D}_{\text{add}}$ might deteriorate the quality of the 5k-pairs seed dictionary due to potentially noisy input.

[6] We also experimented with XLM-R$_{\text{base}}$, but substantially higher overall results were obtained with mBERT as the underlying/input multilingual LM. We plan to analyse these implications in more detail in future work.

[7] Technical details of the learning procedure are described in Appendix A.3. It is important to note that in this case we do not use word translation pairs $(w_{m_i}^x, w_{n_i}^y)$ directly to learn the mapping, but rather each word $w_{m_i}^x$ and $w_{n_i}^y$ is duplicated to create training pairs $(w_{m_i}^x, w_{m_i}^x)$ and $(w_{n_i}^y, w_{n_i}^y)$, where the left word/item in each pair is assigned its WE from C1, and the right word/item is assigned its WE after C2.

Unless noted otherwise, a final representation of an input word w is then a linear combination of (a) its C1-based vector $\mathbf{v}_w$ mapped to a 768-dim representation via $\boldsymbol{W}$, and (b) its 768-dim encoding $f_{\theta'}(w)$ from BLI-tuned mBERT:

$$(1-\lambda)\frac{\mathbf{v}_w\boldsymbol{W}}{\|\mathbf{v}_w\boldsymbol{W}\|_2}+\lambda\frac{f_{\theta'}(w)}{\|f_{\theta'}(w)\|_2}, \quad (7)$$

where λ is a tunable interpolation hyper-parameter.

3 Experimental Setup

Monolingual WEs and BLI Setup. We largely follow the standard BLI setup from prior work (Artetxe et al., 2018; Joulin et al., 2018; Glavaš et al., 2019; Karan et al., 2020, *inter alia*). The main evaluation is based on the standard BLI dataset from Glavaš et al. (2019): it comprises 28 language pairs with a good balance of typologically similar and distant languages: English (EN), German (DE), Italian (IT), French (FR), Russian (RU), Croatian (HR), Turkish (TR), and Finnish (FI). Again following prior work, we rely on monolingual fastText vectors trained on full Wikipedias for each language (Bojanowski et al., 2017), where vocabularies in each language are trimmed to the 200K most frequent words (i.e., $|\mathcal{X}| = 200k$ and $|\mathcal{Y}| = 200k$). The same fastText WEs are used for our Stage C1 and in all baseline BLI models. mBERT in Stage C2 operates over the same vocabularies spanning $200k$ word types in each language.

We use 1k translation pairs (semi-supervised BLI mode) or 5k pairs (supervised) as seed dictionary $\mathcal{D}_0$; test sets span $2k$ pairs (Glavaš et al., 2019). With 56 BLI directions in total,[8] this yields a total of 112 BLI setups for each model in our comparison. The standard *Precision@1* (P@1) BLI measure is reported, and we rely on CSLS (k=10) to score word similarity (Lample et al., 2018).[9]

Training Setup and Hyperparameters. Since standard BLI datasets typically lack a validation set (Ruder et al., 2019), following prior work (Glavaš et al., 2019; Karan et al., 2020) we conduct hyperparameter tuning on a *single, randomly selected* language pair EN→TR, and apply those hyperparameter values in all other BLI runs.

In Stage C1, when $|\mathcal{D}_0|$=$5k$, the hyperparameter values are N_{iter}=2, N_{CL}=200, N_{neg}=150, N_{freq}=$60k$, N_{aug}=$10k$. SGD optimiser is used, with a learning rate of 1.5 and γ=0.99. When $|\mathcal{D}_0|$=$1k$, the values are N_{iter}=3, N_{CL}=50, N_{neg}=60, N_{freq}=$20k$, and N_{aug}=$6k$; SGD with a learning rate of 2.0, γ=1.0. τ=1.0 and dropout is 0 in both cases, and the batch size for contrastive learning is always equal to the size of the current dictionary $|\mathcal{D}_{\text{CL}}|$ (i.e., $|\mathcal{D}_0|$ (5k case), or $|\mathcal{D}_0 \cup \mathcal{D}_{\text{add}}|$ which varies over iterations; see §2.1). In Stage C2, N_{neg}=28 and the maximum sequence length is 6. We use AdamW (Loshchilov and Hutter, 2019) as the optimiser with learning rate of $2e-5$ and weight decay of 0.01. We fine-tune mBERT for 5 epochs, with a batch size of 100; dropout rate is 0.1 and τ=0.1. Unless noted otherwise, λ is fixed to 0.2.

Baseline Models. Our BLI method is evaluated against four strong SotA BLI models from recent literature, all of them with publicly available implementations. Here, we provide brief summaries:[10]

RCSLS (Joulin et al., 2018) optimises a relaxed CSLS loss, learns a non-orthogonal mapping, and has been established as a strong BLI model in empirical comparative analyses as its objective function is directly 'BLI-oriented' (Glavaš et al., 2019).

VecMap's core components (Artetxe et al., 2018) have been outlined in §2.1.

LNMap (Mohiuddin et al., 2020) non-linearly maps the original static WEs into two latent semantic spaces learned via non-linear autoencoders,[11] and then learns another non-linear mapping between the latent autoencoder-based spaces.

FIPP (Sachidananda et al., 2021), in brief, first finds common (i.e., isomorphic) geometric structures in monolingual WE spaces of both languages, and then aligns the Gram matrices of the WEs found in those common structures.

For all baselines, we have verified that the hyperparameter values suggested in their respective repositories yield (near-)optimal BLI performance. Unless noted otherwise, we run VecMap, LNMap, and FIPP with their own self-learning procedures.[12]

[8] For any two languages L_i and L_j, we run experiments both for $L_i \to L_j$ and $L_j \to L_i$ directions.

[9] The same trends in results are observed with Mean Reciprocal Rank (MRR) as another BLI evaluation measure (Glavaš et al., 2019); we omit MRR scores for clarity. Moreover, similar relative trends, but with slightly lower absolute BLI scores, are observed when replacing CSLS with the simpler cosine similarity measure: the results are available in the Appendix.

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

[11] This step is directed towards mitigating anisomorphism (Søgaard et al., 2018; Dubossarsky et al., 2020) between the original WE spaces, which should facilitate their alignment.

[12] RCSLS is packaged without self-learning; extending it to support self-learning is non-trivial and goes beyond the scope of this work.

Model Variants. We denote the full two-stage BLI model as **C2 (Mod)**, where **Mod** refers to the actual model/method used to derive the shared cross-lingual space used by Stage C2. For instance, **C2 (C1)** refers to the model variant which relies on our Stage C1, while **C2 (RCSLS)** relies on RCSLS as the base method. We also evaluate BLI performance of our Stage **C1** BLI method alone.

4 Results and Discussion

The main results are provided in Table 1, while the full results per each individual language pair, and also with cosine similarity as the word retrieval function, are provided in Appendix D. The main findings are discussed in what follows.

Stage C1 versus Baselines. First, we note that there is not a single strongest baseline among the four SotA BLI methods. For instance, RCSLS and VecMap are slightly better than LNMap and FIPP with 5k supervision pairs, while FIPP and VecMap come forth as the stronger baselines with 1k supervision. There are some score fluctuations over individual language pairs, but the average performance of all baseline models is within a relatively narrow interval: the average performance of all four baselines is within 3 P@1 points with 5k pairs (i.e., ranging from 38.22 to 41.22), and VecMap, FIPP, and LNMap are within 2 points with 1k pairs.

Strikingly, contrastive learning in Stage C1 already yields substantial gains over all four SotA BLI models, which is typically much higher than the detected variations between the baselines. We mark that C1 improves over all baselines in 51/56 BLI setups (in the 5k case), and in all 56/56 BLI setups when D_0 spans 1k pairs. The average gains with the C1 variant are ≈5 P@1 points over the strongest baseline in both cases. Note that all the models in comparison, all currently considered SotA in the BLI task, use exactly the same monolingual WEs and leverage exactly the same amount of bilingual supervision. The gains achieved with our Stage C1 thus strongly indicate the potential and usefulness of word-level contrastive fine-tuning when learning linear cross-lingual maps with static WEs (see RQ1 from §1).

Stage C1 + Stage C2. The scores improve further with the full two-stage procedure. The *C2 (C1)* BLI variant increases average P@1 for another 3.3 (5k) and 3 P@1 points (1k), and we observe gains for all language pairs in both translation directions, rendering Stage C2 universally useful. These gains

[5k] Pairs	RCSLS^{+}	VecMap^{x}	LNMap	FIPP	C1	C2 (C1)
DE→∗	43.77	40.49	40.35	40.95	46.14	**48.86**
∗→DE	44.74	42.18	39.55	41.66	46.39	**50.12**
EN→∗	50.94	45.43	44.74	45.76	51.31	**54.31**
∗→EN	49.17	50.19	44.32	47.96	52.61	**55.47**
FI→∗	35.11	36.29	33.18	34.83	39.80	**43.44**
∗→FI	33.49	33.40	34.15	33.00	38.82	**41.97**
FR→∗	47.02	44.67	42.80	44.03	49.12	**51.91**
∗→FR	49.42	48.86	46.25	48.08	51.84	**54.53**
HR→∗	34.06	36.26	33.41	33.52	40.22	**45.53**
∗→HR	32.80	32.96	31.34	31.52	37.82	**42.65**
IT→∗	46.59	44.77	43.23	44.11	48.92	**51.91**
∗→IT	48.41	47.85	45.53	46.64	50.99	**53.85**
RU→∗	40.99	41.01	37.94	39.72	44.17	**47.24**
∗→RU	40.10	35.62	35.66	36.03	42.15	**45.20**
TR→∗	31.29	31.54	30.14	30.34	36.61	**39.86**
∗→TR	31.66	29.42	28.99	28.37	35.67	**39.26**
Avg.	41.22	40.06	38.22	39.16	44.54	**47.88**
[1k] Pairs	RCSLS^{+}	VecMap^{x}	LNMap	FIPP	C1	C2 (C1)
DE→∗	33.43	36.69	37.28	37.70	43.94	**46.61**
∗→DE	32.23	38.63	36.74	39.47	43.15	**46.01**
EN→∗	38.16	38.63	40.44	42.26	47.16	**49.84**
∗→EN	38.57	48.39	43.61	46.68	51.59	**54.03**
FI→∗	22.49	33.08	30.00	32.11	36.81	**40.28**
∗→FI	22.29	27.40	29.95	29.88	36.61	**39.63**
FR→∗	34.98	38.65	39.77	41.08	46.23	**48.57**
∗→FR	36.83	46.61	43.81	46.26	49.75	**52.17**
HR→∗	21.59	33.22	30.05	30.93	37.28	**42.16**
∗→HR	20.87	28.15	27.67	28.15	34.00	**38.77**
IT→∗	36.67	39.45	39.93	42.20	46.55	**49.22**
∗→IT	38.33	45.49	43.47	45.17	48.50	**50.94**
RU→∗	28.45	37.75	35.13	38.24	42.21	**44.61**
∗→RU	27.78	26.16	29.71	31.28	38.02	**41.04**
TR→∗	18.72	26.97	26.63	27.05	33.77	**36.89**
∗→TR	17.59	23.63	24.26	24.68	32.34	**35.57**
Avg.	29.31	35.56	34.90	36.45	41.74	**44.77**

Table 1: P@1 scores on the BLI benchmark of Glavaš et al. (2019) with bilingual supervision (i.e., $\mathcal{D}_0$ size) of 5k (upper half) and 1k translation pairs (bottom half). $L \rightarrow *$ and $* \rightarrow L$ denote the average BLI scores of BLI setups where L is the source and the target language, respectively. The word similarity measure is CSLS (see §3). Underlined scores are the peak scores among methods that rely solely on static fastText WEs; **Bold** scores denote the highest scores overall (i.e., the use of word translation knowledge exposed from mBERT is allowed). $^{+}$RCSLS is always used without self learning (see the footnote in 3); ^{x}We report VecMap with self-learning in the 1k-pairs scenario, and its variant without self-learning when using supervision of 5k pairs as it performs better than the variant with self-learning.

indicate that mBERT does contain word translation knowledge in its parameters. However, the model must be fine-tuned (i.e., transformed) to 'unlock' the knowledge from its parameters: this is done through a BLI-guided contrastive fine-tuning procedure (see §2.2). Our findings thus further confirm the 'rewiring hypothesis' from prior work (Vulić et al., 2021; Liu et al., 2021b; Gao et al., 2021), here validated for the BLI task (see RQ2 from §1), which states that task-relevant knowledge at sentence- and word-level can be 'rewired'/exposed from the off-the-shelf LMs, even when leveraging very limited task supervision, e.g., with only 1k or 5k word translation pairs as in our experiments.

[1k] **Pairs**	**BG→CA**	**CA→HE**	**HE→BG**
VecMap	39.43	24.64	31.55
FIPP	34.29	20.63	26.38
C1	41.88	30.56	33.49
C2 (C1)	**44.28**	**33.99**	**37.78**

Table 2: BLI scores on the Panlex-BLI sets.

[5k] **Pairs**	**DE→TR**	**TR→HR**	**HR→RU**
RCSLS	30.99	24.60	37.19
C2 (RCSLS)	36.52	33.17	44.77
VecMap	27.18	25.99	37.98
C2 (VecMap)	34.95	34.29	44.98
C1	34.69	32.37	41.66
C2 (C1)	**38.86**	**36.32**	**46.40**
[1k] **Pairs**	**DE→TR**	**TR→HR**	**HR→RU**
RCSLS	18.21	13.84	24.72
C2 (RCSLS)	25.40	22.52	33.88
VecMap	23.37	20.50	36.09
C2 (VecMap)	27.91	26.84	40.45
C1	32.03	27.00	39.40
C2 (C1)	**34.85**	**32.16**	**42.14**

Table 3: Stage C2 with different 'support' methods: RCSLS, VecMap, and C1. P@1×100% scores.

[1k] **Pairs**	**EN→∗**	**DE→∗**	**IT→∗**
C1 w/o CL	39.46	37.54	40.37
C1 w/o SL	39.31	32.59	36.45
C1	47.16	43.94	46.55
mBERT	9.55	9.39	8.13
mBERT (tuned)	17.29	20.92	23.29
C1 + mBERT	47.56	44.08	46.74
C2 (C1)	**49.84**	**46.61**	**49.22**

Table 4: Ablation study. CL = Contrastive Learning; SL = Self-Learning. 'mBERT' and 'mBERT (tuned)' refer to using word encodings from mBERT directly for BLI, before and after fine-tuning in Stage C2. Very similar trends are observed for all other language pairs (available in Appendix F).

Performance over Languages. The absolute BLI scores naturally depend on the actual source and target languages: e.g., the lowest absolute performance is observed for morphologically rich (HR, RU, FI, TR) and non-Indo-European languages (FI, TR). However, both C1 and C2 (C1) mode variants offer wide and substantial gains in performance for *all* language pairs, irrespective of the starting absolute score. This result further suggests wide applicability and robustness of our BLI method.

4.1 Further Discussion

Evaluation on Lower-Resource Languages. The robustness of our BLI method is further tested on another BLI evaluation set: PanLex-BLI (Vulić et al., 2019), which focuses on BLI evaluation for lower-resource language; 1k training pairs and 2k test pairs are derived from PanLex (Kamholz et al., 2014). The results for a subset of three languages (Bulgarian: BG, Catalan: CA, Hebrew: HE) are presened in Table 2, with more results available in Appendix E. Overall, the results further confirm the efficacy of the *C2 (C1)*, with gains observed even with typologically distant language pairs (e.g., CA→HE and HE→BG).

Usefulness of Stage C2? The results in Table 1 have confirmed the effectiveness of our two-stage *C2 (C1)* BLI method (see RQ3 in §1). However, Stage C2 is in fact independent of our Stage C1, and thus can also be combined with other standard BLI methods. Therefore, we seek to validate whether combining exposed mBERT-based translation knowledge can also aid other BLI methods. In other words, instead of drawing positive and negative samples from Stage C1 (§2.2) and combining C2 WEs with WEs from C1 (§2.3), we replace C1 with our baseline models. The results of these *C2 (RCSLS)* and *C2 (VecMap)* BLI variants for a selection of language pairs are provided in Table 3.

The gains achieved with all *C2 (·)* variants clearly indicate that Stage C2 produces WEs which aid all BLI methods. In fact, combining it with RCSLS and VecMap yields even larger relative gains over the base models than combining it with our Stage C1. However, since Stage C1 (as the base model) performs better than RCSLS and VecMap, the final absolute scores with *C2 (C1)* still outperform *C2 (RCSLS)* and *C2 (VecMap)*.

Combining C1 and C2? The usefulness of combining the representations from two stages is measured through varying the value of λ for several BLI setups. The plots are shown in Figure 2, and indicate that Stage C1 is more beneficial to the performance, with slight gains achieved when allowing the 'influx' of mBERT knowledge (e.g., λ in the $[0.0 - 0.3]$ interval). While mBERT-based WEs are not sufficient as standalone representations for BLI, they seem to be even more useful in the combined model for lower-resource languages on PanLex-BLI, with steeper increase in performance, and peak scores achieved with larger λ-s.

Ablation Study, with results summarised in Table 4, displays several interesting trends. First, both CL and self-learning are key components in the 1k-setups: removing any of them yields substantial drops.[13] Further, Table 4 complements the results from Figure 2 and again indicates that, while Stage C2 indeed boosts word translation capacity

[13] In 5k-setups, self-learning becomes less important, and removing it yields only negligible drops, while contrastive self-tuning remains a crucial component, see the Appendix.

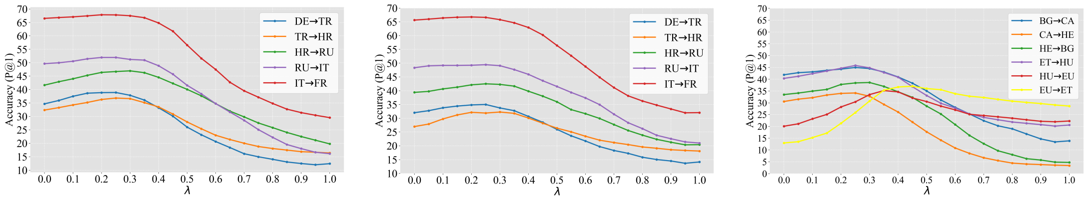

Figure 2: BLI scores with different λ values: (left) $|\mathcal{D}_0|$=5k; (middle) $|\mathcal{D}_0|$=1k; (right) PanLex-BLI, $|\mathcal{D}_0|$=1k.

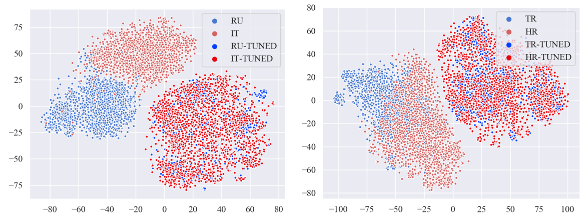

Figure 3: t-SNE visualisation (van der Maaten and Hinton, 2012) of mBERT encodings of words from BLI test sets for RU-IT (left) and TR-HR (right). Similar plots for more language pairs are in Appendix C.

of mBERT, using mBERT features alone is still not sufficient to achieve competitive BLI scores. Finally, Table 4 shows the importance of fine-tuning mBERT before combining it with C1-based WEs (§2.3): directly adding WEs extracted from the off-the-shelf mBERT does not yield any benefits (see the scores for the *C1+mBERT* variant).

The impact of contrastive fine-tuning on mBERT's representation space for two language pairs is illustrated by a t-SNE plot in Figure 3. The semantic space of off-the-shelf mBERT displays a clear separation of language-specific subspaces (Libovický et al., 2020; Dufter and Schütze, 2020), which makes it unsuitable for the BLI task. On the other hand, contrastive fine-tuning reshapes the subspaces towards a shared (cross-lingual) space, the effects of which are then also reflected in mBERT's improved BLI capability (see Table 4 again).

5 Related Work

This work is related to three topics, each with a large body of work; we can thus provide only a condensed summary of the most relevant research.

Mapping-based BLI. These BLI methods are highly popular due to reduced bilingual supervision requirements; consequently, they are applicable to low-resource languages and domains, learning linear (Lample et al., 2018; Artetxe et al., 2018; Joulin et al., 2018; Patra et al., 2019; Jawanpuria et al., 2019; Sachidananda et al., 2021) and non-linear maps (Mohiuddin et al., 2020; Glavaš and Vulić, 2020; Ganesan et al., 2021), typically using self-learning in weakly supervised setups.

Contrastive Learning in NLP aims to learn a semantic space such that embeddings of similar text inputs are close to each other, while 'repelling' dissimilar ones. It has shown promising performance on training generic sentence encoders (Giorgi et al., 2021; Carlsson et al., 2021; Liu et al., 2021a; Gao et al., 2021) and downstream tasks like summarisation (Liu and Liu, 2021) or NER (Das et al., 2021).

Exposing Lexical Knowledge from Pretrained LMs. Extracting lexical features from off-the-shelf multilingual LMs typically yields subpar performance in lexical tasks (Vulić et al., 2020b). To unlock the lexical knowledge encoded in PLMs, Liu et al. (2021a); Vulić et al. (2021) fine-tune LMs via contrastive learning with manually curated or automatically extracted phrase/word pairs to transform it into effective text encoders. Wang et al. (2021) and Liu et al. (2021c) apply similar techniques for phrase and word-in-context representation learning respectively. The success of these methods suggests that LMs store a wealth of lexical knowledge: yet, as we confirm here for BLI, fine-tuning is typically needed to expose this knowledge.

6 Conclusion

We have proposed a simple yet extremely effective and robust two-stage contrastive learning framework for improving bilingual lexicon induction (BLI). In Stage C1, we tune cross-lingual linear mappings between static word embeddings with a contrastive objective and achieve substantial gains in 107 out of 112 BLI setups on the standard BLI benchmark. In Stage C2, we further propose a contrastive fine-tuning procedure to harvest cross-lingual lexical knowledge from multilingual pretrained language models. The representations from this process, when combined with Stage C1 embeddings, have resulted in further boosts in BLI performance, with large gains in all 112 setups. We have also conducted a series of finer-grained evaluations, analyses and ablation studies.

References

Ekin Akyurek and Jacob Andreas. 2021. Lexicon learning for few shot sequence modeling. 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 4934–4946, Online. Association for Computational Linguistics.

Alan Ansell, Edoardo Maria Ponti, Anna Korhonen, and Ivan Vulić. 2021. Composable sparse fine-tuning for cross-lingual transfer. *CoRR*, abs/2110.07560.

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.

Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2021. Semantic re-tuning with contrastive tension. In *Proceedings of the International Conference on Learning Representations (ICLR'21)*.

Ethan C. Chau and Noah A. Smith. 2021. Specializing multilingual language models: An empirical study. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 51–61, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alexandra Chronopoulou, Dario Stojanovski, and Alexander Fraser. 2021. Improving the lexical ability of pretrained language models for unsupervised neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'21)*, pages 173–180, Online. Association for Computational Linguistics.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca J Passonneau, and Rui Zhang. 2021. Container: Few-shot named entity recognition via contrastive learning. *arXiv preprint arXiv:2109.07589*.

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.

Sumanth Doddapaneni, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2021. A primer on pretrained multilingual language models. *CoRR*, abs/2107.00676.

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.

Haim Dubossarsky, Ivan Vulić, Roi Reichart, and Anna Korhonen. 2020. The secret is in the spectra: Predicting cross-lingual task performance with spectral similarity measures. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*, pages 2377–2390, Online. Association for Computational Linguistics.

Philipp Dufter and Hinrich Schütze. 2020. Identifying elements essential for BERT's multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*, pages 4423–4437, Online. Association for Computational Linguistics.

Ashwinkumar Ganesan, Francis Ferraro, and Tim Oates. 2021. Learning a reversible embedding mapping using bi-directional manifold alignment. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3132–3139, Online. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP'21)*, pages 6894–6910, Online and Punta Cana, Dominican Republic. 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.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association* *for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP'21)*, pages 879–895, Online. Association for Computational Linguistics.

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.

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.

Yedid Hoshen and Lior Wolf. 2018. Non-adversarial unsupervised word translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*, pages 469–478, Brussels, Belgium. Association for Computational Linguistics.

Ann Irvine and Chris Callison-Burch. 2017. A comprehensive analysis of bilingual lexicon induction. *Computational Linguistics*, 43(2):273–310.

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.

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)*.

Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. 2020. On the language neutrality of pre-trained multilingual representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1663–1674, Online. Association for Computational Linguistics.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021a. Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'21)*, pages 4228–4238, Online. Association for Computational Linguistics.

Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2021b. Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP'21)*, pages 1442–1459, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qianchu Liu, Fangyu Liu, Nigel Collier, Anna Korhonen, and Ivan Vulić. 2021c. MirrorWiC: On eliciting word-in-context representations from pretrained language models. In *Proceedings of the 25th Conference on Computational Natural Language Learning (CoNLL'21)*, pages 562–574, Online. Association for Computational Linguistics.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. 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 1065–1072, Online. Association for Computational Linguistics.

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. *ArXiv preprint*, abs/1309.4168.

Tasnim Mohiuddin, M Saiful Bari, and Shafiq Joty. 2020. LNMap: Departures from isomorphic assumption in bilingual lexicon induction through nonlinear 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.

Tasnim Mohiuddin and Shafiq Joty. 2019. Revisiting adversarial autoencoder for unsupervised word translation with cycle consistency and improved training. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'19)*, pages 3857–3867, Minneapolis, Minnesota. Association for Computational Linguistics.

Kevin Musgrave, Serge J. Belongie, and Ser-Nam Lim. 2020. A metric learning reality check. In *Proceedings of the European Conference on Computer Vision (ECCV'20)*, pages 681–699.

Arbi Haza Nasution, Yohei Murakami, and Toru Ishida. 2021. Plan optimization to bilingual dictionary induction for low-resource language families. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 20(2):29:1–29:28.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *ArXiv preprint*, abs/1807.03748.

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.

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.

Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *33rd Annual Meeting of the Association for Computational Linguistics*, 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.

Shuo Ren, Shujie Liu, Ming Zhou, and Shuai Ma. 2020. A graph-based coarse-to-fine method for unsupervised bilingual lexicon induction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*, pages 3476–3485, Online. 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)*.

Peter H Schönemann. 1966. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31(1):1–10.

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.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18)*, pages 778–788, Melbourne, Australia. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey E. Hinton. 2012. Visualizing non-metric similarities in multiple maps. *Machine Learning*, 87(1):33–55.

Thomas Viklands. 2006. *Algorithms for the weighted orthogonal Procrustes problem and other least squares problems*. Ph.D. thesis, Datavetenskap.

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, 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.

Shufan Wang, Laure Thompson, and Mohit Iyyer. 2021. Phrase-BERT: Improved phrase embeddings from BERT with an application to corpus exploration. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10837–10851, Online and Punta Cana, Dominican Republic. 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 Technical Details and Further Clarifications

A.1 Advanced Mapping (AM) in Stage C1

Suppose $\boldsymbol{X}_{\mathcal{D}}, \boldsymbol{Y}_{\mathcal{D}} \in \mathcal{R}^{|\mathcal{D}|\times d}$ are source and target embedding matrices corresponding to the training dictionary $\mathcal{D}$. Then $\boldsymbol{X}_{\mathcal{D}}^T$ and $\boldsymbol{Y}_{\mathcal{D}}^T$ are whitened, and singular value decomposition (SVD) is conducted on the whitened embeddings:

$$\boldsymbol{X}_{\mathcal{D}}^{'} = \boldsymbol{X}_{\mathcal{D}}(\boldsymbol{X}_{\mathcal{D}}^T\boldsymbol{X}_{\mathcal{D}})^{-\frac{1}{2}}, \tag{8}$$

$$\boldsymbol{Y}_{\mathcal{D}}^{'} = \boldsymbol{Y}_{\mathcal{D}}(\boldsymbol{Y}_{\mathcal{D}}^T\boldsymbol{Y}_{\mathcal{D}})^{-\frac{1}{2}}, \tag{9}$$

$$\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T = \boldsymbol{X}_{\mathcal{D}}^{'T}\boldsymbol{Y}_{\mathcal{D}}^{'}. \tag{10}$$

$\boldsymbol{W}_x$ and $\boldsymbol{W}_y$ are then derived after re-weighting and de-whitening as follows:

$$\boldsymbol{W}_x = (\boldsymbol{X}_{\mathcal{D}}^T\boldsymbol{X}_{\mathcal{D}})^{-\frac{1}{2}}\boldsymbol{U}\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{U}^T(\boldsymbol{X}_{\mathcal{D}}^T\boldsymbol{X}_{\mathcal{D}})^{\frac{1}{2}}\boldsymbol{U}, \tag{11}$$

$$\boldsymbol{W}_y = (\boldsymbol{Y}_{\mathcal{D}}^T\boldsymbol{Y}_{\mathcal{D}})^{-\frac{1}{2}}\boldsymbol{V}\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{V}^T(\boldsymbol{Y}_{\mathcal{D}}^T\boldsymbol{Y}_{\mathcal{D}})^{\frac{1}{2}}\boldsymbol{V}. \tag{12}$$

A.2 Word Similarity/Retrieval Measures

Given two word embeddings $\mathbf{x} \in \boldsymbol{X}$ and $\mathbf{y} \in \boldsymbol{Y}$, their similarity can be defined as their cosine similarity $m(\mathbf{x}, \mathbf{y}) = \text{cosine}(\mathbf{x}, \mathbf{y})$. In the FIPP model, we calculate dot product $m(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \cdot \mathbf{y}$ between $\mathbf{x}$ and $\mathbf{y}$ instead without normalisation, as with FIPP this produces better BLI scores in general. [14].

For the simple Nearest Neighbor (NN) BLI with cosine (or dot product), we retrieve the word from the entire target language vocabulary of size 200k with the highest similarity score and mark it as the translation of the input/query word in the source language.

For the Cross-domain Similarity Local Scaling (CSLS) measure, a CSLS score is defined as $\text{CSLS}(\mathbf{x}, \mathbf{y}) = 2m(\mathbf{x}, \mathbf{y}) - r_{\boldsymbol{X}}(\mathbf{y}) - r_{\boldsymbol{Y}}(\mathbf{x})$. $r_{\boldsymbol{X}}(y)$ is the average $m(\cdot,\cdot)$ score of $\mathbf{y}$ and its k-NNs ($k = 10$) in $\boldsymbol{X}$; $r_{\boldsymbol{Y}}(\mathbf{x})$ is the average $m(\cdot,\cdot)$ scores of x and its k-NNs ($k = 10$) in $\boldsymbol{Y}$. Note that when using CSLS scores to retrieve the translation of $\mathbf{x}$ in $\boldsymbol{Y}$, the term $r_{\boldsymbol{Y}}(\mathbf{x})$ can be ignored, as it is a constant for all $\mathbf{y}$, and we can similarly ignore $r_{\boldsymbol{X}}(\mathbf{y})$ when doing BLI in the opposite direction.

[14] https://github.com/vinsachi/FIPPCLE/blob/main/xling-bli/code/eval.py

A.3 Generalised Procrustes in Stage C2

We consider the following Procrustes problem:

$$\underset{\boldsymbol{W}}{\text{argmin}} \, \|\boldsymbol{X}\boldsymbol{W} - \boldsymbol{Y}\|_F^2, \boldsymbol{W}\boldsymbol{W}^T = \boldsymbol{I}, \tag{13}$$

where $\boldsymbol{X} \in \mathbb{R}^{n\times d_1}$ is a C1-induced cross-lingual space spanning all source and target words in the training set $\mathcal{D}$, $\boldsymbol{Y} \in \mathbb{R}^{n\times d_2}$ is a C2-induced space representing all mBERT-encoded vectors corresponding to the same words from $\boldsymbol{X}$, and $\boldsymbol{W} \in \mathbb{R}^{d_1\times d_2}$, $d_1 \leq d_2$. A classical Orthogonal Procrustes Problem assumes that $d_1 = d_2$ and $\boldsymbol{W}$ is an orthogonal matrix (i.e., it should be a square matrix), where its optimal solution is given by $\boldsymbol{U}\boldsymbol{V}^T$; here, $\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$ is the full singular value decomposition (SVD) of $\boldsymbol{X}^T\boldsymbol{Y}$. In our experiments, we need to address the case $d_1 < d_2$ when mapping 300-dimensional static fastText WEs to the 768-dimensional space of mBERT-based WEs. It is easy to show that when $d_1 < d_2$, $\boldsymbol{U}[\boldsymbol{S}, \boldsymbol{0}]\boldsymbol{V}^T$=$\boldsymbol{X}^T\boldsymbol{Y}$ (again the full SVD decomposition), the optimal $\boldsymbol{W}$ is then $\boldsymbol{U}[\boldsymbol{I}, \boldsymbol{0}]\boldsymbol{V}^T$ (it degrades to the Orthogonal Procrustes Problem when $d_1 = d_2$). Below, we provide a simple proof.

Let $\boldsymbol{\Omega}$=$\boldsymbol{U}^T\boldsymbol{W}\boldsymbol{V}$, then $\boldsymbol{\Omega}\boldsymbol{\Omega}^T = \boldsymbol{I}$. Therefore, each of its element $-1 \leq \boldsymbol{\Omega}_{i,j} \leq 1$.

$$\begin{aligned}
&\underset{\boldsymbol{W}}{\text{argmin}} \, \|\boldsymbol{X}\boldsymbol{W} - \boldsymbol{Y}\|_F^2 \\
=&\underset{\boldsymbol{W}}{\text{argmin}} \langle \boldsymbol{X}\boldsymbol{W} - \boldsymbol{Y}, \boldsymbol{X}\boldsymbol{W} - \boldsymbol{Y}\rangle_F \\
=&\underset{\boldsymbol{W}}{\text{argmin}} \, \|\boldsymbol{X}\boldsymbol{W}\|_F^2 + \|\boldsymbol{Y}\|_F^2 - 2\langle \boldsymbol{X}\boldsymbol{W}, \boldsymbol{Y}\rangle_F \\
=&\underset{\boldsymbol{W}}{\text{argmax}} \langle \boldsymbol{X}\boldsymbol{W}, \boldsymbol{Y}\rangle_F \\
=&\underset{\boldsymbol{W}}{\text{argmax}} \langle \boldsymbol{W}, \boldsymbol{X}^T\boldsymbol{Y}\rangle_F \\
=&\underset{\boldsymbol{W}}{\text{argmax}} \langle \boldsymbol{W}, \boldsymbol{X}^T\boldsymbol{Y}\rangle_F \\
=&\underset{\boldsymbol{W}}{\text{argmax}} \langle \boldsymbol{W}, \boldsymbol{U}[\boldsymbol{S}, \boldsymbol{0}]\boldsymbol{V}^T\rangle_F \\
=&\underset{\boldsymbol{W}}{\text{argmax}} \langle [\boldsymbol{S}, \boldsymbol{0}], \boldsymbol{U}^T\boldsymbol{W}\boldsymbol{V}\rangle_F \\
=&\underset{\boldsymbol{W}}{\text{argmax}} \langle [\boldsymbol{S}, \boldsymbol{0}], \boldsymbol{\Omega}\rangle_F
\end{aligned} \tag{14}$$

In the formula above, $\|\cdot\|_F$ and $\langle\cdot,\cdot\rangle_F$ are Frobenius norm and Frobenius inner product, and we leverage their properties throughout the proof. Note that S is a diagonal matrix with non-negative elements and thus the maximum is achieved when $\boldsymbol{\Omega}$=$[\boldsymbol{I}, \boldsymbol{0}]$ and $\boldsymbol{W}$=$\boldsymbol{U}[\boldsymbol{I}, \boldsymbol{0}]\boldsymbol{V}^T$.

Note that the Procrustes mapping over word embedding matrices keeps word similarities on both sides intact. Since $\boldsymbol{W}\boldsymbol{W}^T{=}\boldsymbol{I}$, $\cos(\mathbf{x}_i\boldsymbol{W}, \mathbf{x}_j\boldsymbol{W}){=}\cos(\mathbf{x}_i, \mathbf{x}_j)$.

We would also like to add an additional note, although irrelevant to our own experiments, that the above derivation cannot address $d_1 > d_2$ scenarios: in that case $\boldsymbol{W}\boldsymbol{W}^T$ cannot be a full-rank matrix and thus $\boldsymbol{W}\boldsymbol{W}^T \neq \boldsymbol{I}$.

A.4 Languages in BLI Evaluation

	Language	Family	Code
XLING	Croatian	Slavic	HR
	English	Germanic	EN
	Finnish	Uralic	FI
	French	Romance	FR
	German	Germanic	DE
	Italian	Romance	IT
	Russian	Slavic	RU
	Turkish	Turkic	TR
PanLex-BLI	Basque	–(isolate)	EU
	Bulgarian	Slavic	BG
	Catalan	Romance	CA
	Estonian	Uralic	ET
	Hebrew	Afro-Asiatic	HE
	Hungarian	Uralic	HU

Table 5: A list of languages in our experiments along with their language family and ISO 639-1 code.

B Reproducibility Checklist

- **BLI Data**: The two BLI datasets are publicly available.[15] [16]
- **Static WEs**: We use the preprocessed fastText WEs provided by Glavaš et al. (2019). For PanLex-BLI, we follow the the original paper's setup (Vulić et al., 2019) and adopt fastText WEs pretrained on both Common Crawl and Wikipedia(Bojanowski et al., 2017).[17] Following prior work, all static WEs are trimmed to contain vectors for the top 200k most frequent words in each language.
- **Baseline BLI Models**: All models are accessible online as publicly available github repositories.
- **Pretrained LM**: The used mBERT variant is 'bert-base-multilingual-uncased', retrieved from the huggingface.co model repository.
- **Source Code**: Our code is available online at: `[URL-ANONYMOUS]`.
- **Computing Infrastructure**: We run our code on a machine with a 4.00GHz 4-core i7-6700K CPU, 64GB RAM and two 12GB NVIDIA TITAN X GPUs. We rely on Python 3.6.10, PyTorch 1.7.0 and huggingface.co Transformers 4.4.2. Automatic Mixed Precision (AMP)[18] is leveraged during C2 training.
- **Runtime**: The training process (excluding data loading and evaluation) typically takes 650 seconds for Stage C1 (seed dictionary of 5k, 2 self-learning iterations) and 200 seconds for C1 (1k, 3 self-learning iterations) on a single GPU. Stage C2 runs for $\approx$ 500 seconds on two GPUs.

C Visualisation of mBERT-Based Word Representations

To illustrate the impact of the proposed BLI-oriented fine-tuning of mBERT in Stage C2 on its representation space, we visualise the 768-dimensional mBERT word representations (i.e., mBERT-encoded word features alone, without the infusion of C1-aligned static WEs). We encode BLI test sets (i.e., these sets include 2k source-target word pairs unseen during C2 fine-tuning), before and after fine-tuning, relying on 1k training samples as the seed dictionary D_0.

Here, we provide comparative t-SNE visualisations between source and target word mBERT-based decontextualised word representations (see §2.2) for six language pairs from the BLI dataset of Glavaš et al. (2019): EN-IT, FI-RU, EN-HR, HR-RU, DE-TR, and IT-FR, while two additional visualisations are available in the main paper (for RU-IT and TR-HR, see Figure 3 in §4.1). As visible in all the figures below, BLI-oriented fine-tuning in Stage C2, there is an obvious mismatch between mBERT's representation subspaces in the two languages. This undesired property gets mitigated, to a considerable extent, by the fine-tuning procedure in Stage C2.

[15] https://github.com/vinsachi/FIPPCLE/blob/main/xling-bli/code/eval.py
[16] https://github.com/cambridgeltl/panlex-bli
[17] https://fasttext.cc/docs/en/crawl-vectors.html
[18] https://pytorch.org/docs/stable/amp. html

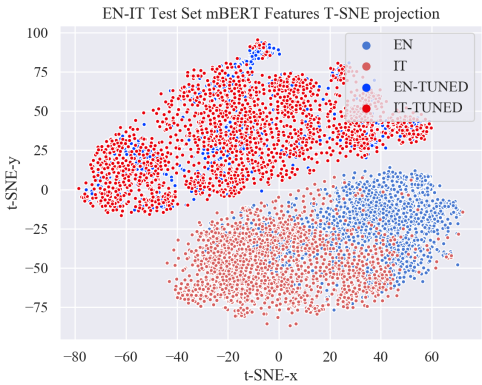

Figure 4: A t-SNE visualisation of mBERT-encoded representations of words from the EN-IT BLI test set. The representations before BLI-oriented fine-tuning of mBERT in Stage C2 are plotted in muted blue and red, and after fine-tuning in bright colours.

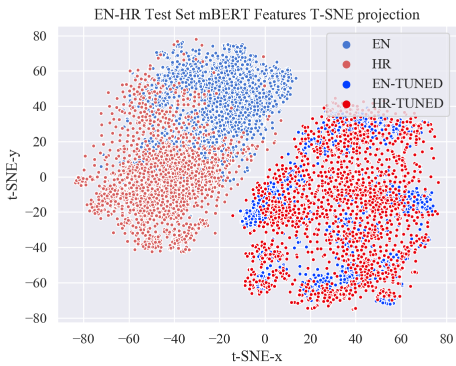

Figure 5: A t-SNE visualisation of mBERT-encoded representations of words from the EN-HR BLI test set. The representations before BLI-oriented fine-tuning of mBERT in Stage C2 are plotted in muted blue and red, and after fine-tuning in bright colours.

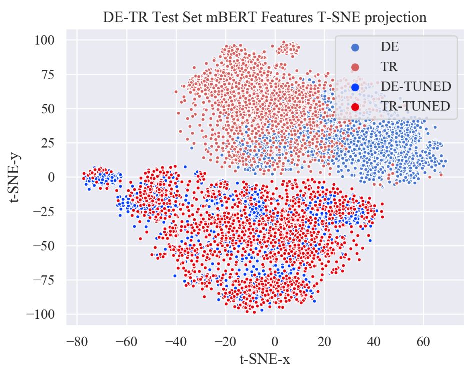

Figure 6: A t-SNE visualisation of mBERT-encoded representations of words from the DE-TR BLI test set. The representations before BLI-oriented fine-tuning of mBERT in Stage C2 are plotted in muted blue and red, and after fine-tuning in bright colours.

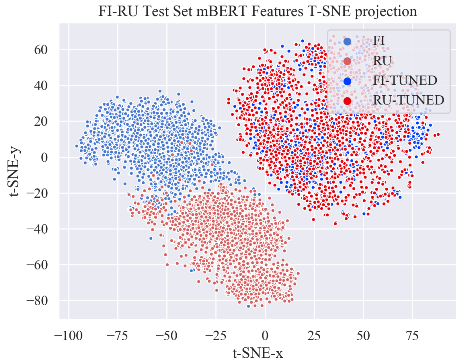

Figure 7: A t-SNE visualisation of mBERT-encoded representations of words from the FI-RU BLI test set. The representations before BLI-oriented fine-tuning of mBERT in Stage C2 are plotted in muted blue and red, and after fine-tuning in bright colours.

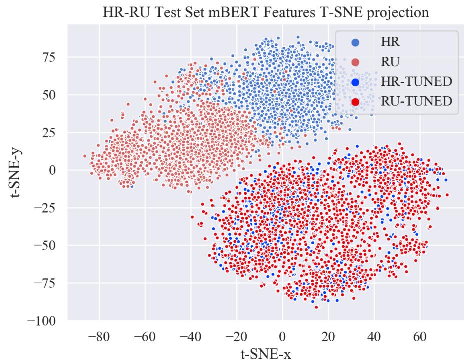

Figure 8: A t-SNE visualisation of mBERT-encoded representations of words from the HR-RU BLI test set. The representations before BLI-oriented fine-tuning of mBERT in Stage C2 are plotted in muted blue and red, and after fine-tuning in bright colours.

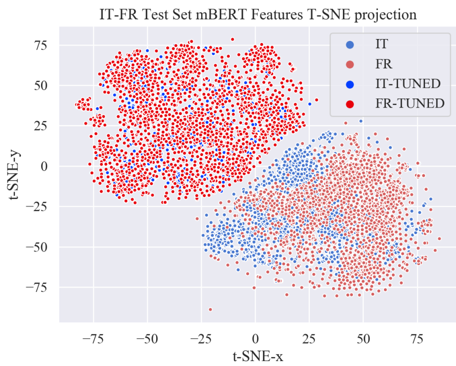

Figure 9: A t-SNE visualisation of mBERT-encoded representations of words from the IT-FR BLI test set. The representations before BLI-oriented fine-tuning of mBERT in Stage C2 are plotted in muted blue and red, and after fine-tuning in bright colours.

D Appendix: Full BLI Results

Complete results on the BLI dataset of Glavaš et al. (2019), per each language pair and also including NN-based BLI scores, are provided in Tables 6-7. It can be seen as an expanded variant of the main Table 1 presented in the main paper.

E Appendix: Additional Results on the PanLex-BLI Evaluation Set

Additional results on the PanLex-BLI evaluation set, focused on typologically diverse and low-resource languages, for a subset of 3 more languages are provided in Table 8. These results are related to the discussion in §4.1 in the main paper.

F Appendix: Full Ablation Study

Complete results of the ablation study, over all languages in the evaluation set of Glavaš et al. (2019), are available in Table 9, and can be seen as additional evidence which supports the claims from the main paper (see §4.1)

[5k] **Pairs**	**RCSLS**	**VecMap-Sup**	**LNMap**	**FIPP**	**C1**	**C2 (C1)**
DE→FI	30.62/37.35	29.21/33.59	31.35/36.10	30.93/35.37	38.97/42.10	**41.47/44.65**
FI→DE	32.48/39.36	35.42/38.73	31.32/36.73	36.05/39.41	39.83/42.46	**44.30/47.03**
DE→FR	47.63/52.74	46.64/50.44	44.91/48.46	47.89/50.44	51.49/53.78	**54.09/55.56**
FR→DE	47.23/51.22	45.37/47.75	41.65/44.80	45.73/47.85	50.13/51.37	**53.23/53.29**
DE→HR	29.26/33.75	27.07/32.08	27.65/32.34	27.65/31.09	34.17/37.66	**39.07/42.41**
HR→DE	30.30/36.35	32.98/37.24	28.98/33.72	31.51/34.30	39.14/41.35	**45.03/48.29**
DE→IT	47.68/52.63	47.78/50.55	44.91/47.94	46.90/49.97	50.65/52.79	**52.48/54.77**
IT→DE	46.51/51.01	44.96/47.29	42.58/45.53	44.86/46.67	49.97/51.21	**53.90/53.80**
DE→RU	37.87/42.41	31.98/34.38	35.21/37.92	36.57/37.09	42.67/44.29	**44.71/46.79**
RU→DE	40.54/45.78	40.65/43.32	36.72/40.28	40.18/42.38	46.05/46.73	**48.51/49.71**
DE→TR	24.93/30.99	23.84/27.18	25.46/29.16	23.94/27.65	31.30/34.69	**35.84/38.86**
TR→DE	27.00/31.84	26.46/29.93	24.92/27.85	26.09/29.18	33.33/36.74	**38.50/40.95**
EN→DE	52.95/57.60	48.65/51.00	45.80/47.95	50.25/51.85	55.50/54.90	**59.25/57.75**
DE→EN	50.97/56.55	52.01/55.24	46.48/50.50	52.16/55.03	54.77/57.69	**56.03/58.95**
EN→FI	35.40/42.05	35.25/37.75	34.45/38.35	34.55/39.10	40.70/44.60	**45.45/47.15**
FI→EN	34.21/41.25	39.04/43.51	31.69/36.26	36.42/40.51	41.46/46.30	**44.82/50.55**
EN→FR	61.65/66.55	60.65/63.10	57.75/62.10	61.15/63.25	64.35/65.05	**68.45/67.20**
FR→EN	59.23/63.11	59.60/62.75	54.53/58.72	59.03/61.87	62.23/63.84	**64.30/65.49**
EN→HR	31.40/37.90	29.70/34.05	28.40/31.75	28.50/31.95	37.50/40.70	**43.60/47.20**
HR→EN	28.51/35.67	35.24/39.08	27.83/32.61	31.93/34.72	38.66/42.40	**42.61/49.08**
EN→IT	58.85/64.05	57.20/60.40	55.30/59.05	56.95/59.75	61.55/63.45	**65.30/65.60**
IT→EN	55.09/61.50	57.73/62.17	52.09/56.02	56.69/60.52	59.90/63.51	**62.27/65.27**
EN→RU	44.75/49.40	38.00/39.65	38.90/41.10	40.70/42.00	48.05/49.15	**50.85/50.50**
RU→EN	42.80/48.66	45.78/49.35	37.51/42.64	43.27/47.15	48.45/51.91	**49.24/54.16**
EN→TR	31.40/39.05	30.35/32.05	29.55/32.85	30.80/32.40	39.10/41.35	**43.55/44.75**
TR→EN	30.78/37.43	34.45/39.24	28.12/33.49	31.79/35.89	39.03/42.60	**39.24/44.78**
FI→FR	30.90/36.73	34.68/38.26	29.16/34.79	33.79/37.26	38.94/42.20	**42.77/45.24**
FR→FI	29.59/34.92	31.35/34.30	30.42/33.26	30.11/33.26	36.42/39.99	**41.18/43.20**
FI→HR	22.65/28.06	27.17/31.58	24.65/29.06	25.54/29.06	30.16/34.89	**34.52/38.31**
HR→FI	18.20/26.35	28.30/31.72	26.67/31.93	25.78/29.30	32.51/35.61	**37.40/39.56**
FI→IT	31.53/36.94	33.89/37.99	31.37/35.58	33.58/36.15	38.47/42.04	**42.51/46.30**
IT→FI	29.56/34.21	31.06/34.32	31.47/35.09	29.97/33.54	35.76/39.48	**40.78/43.57**
FI→RU	28.74/34.52	31.16/34.16	28.38/32.32	30.37/32.79	35.10/37.73	**38.36/40.99**
RU→FI	27.29/33.11	29.91/33.53	28.60/33.63	27.82/32.53	35.57/36.98	**38.55/40.91**
HR→FR	33.46/39.66	35.35/40.24	30.72/36.09	35.30/38.72	39.61/44.13	**45.40/49.29**
FR→HR	30.94/35.28	29.85/33.21	26.90/30.88	29.69/33.26	36.32/39.78	**40.71/44.08**
HR→IT	29.62/37.98	36.24/40.24	32.14/36.72	34.19/36.98	38.93/43.77	**44.71/48.97**
IT→HR	30.34/34.06	30.75/34.32	27.80/32.87	30.03/33.49	37.26/38.71	**41.40/44.75**
HR→RU	31.35/37.19	34.19/37.98	32.40/36.61	33.19/36.03	39.40/41.66	**44.35/46.40**
RU→HR	31.48/35.94	34.57/39.50	31.48/35.78	32.16/36.56	37.93/40.60	**42.17/45.47**
IT→FR	64.19/66.51	64.03/65.89	62.12/64.60	63.57/65.32	65.37/66.51	**66.82/67.86**
FR→IT	62.96/66.11	62.70/64.72	61.05/63.68	62.18/64.30	64.25/66.27	**66.79/67.20**
RU→FR	44.00/47.67	43.58/47.51	38.82/43.64	42.90/47.15	48.04/50.55	**50.13/52.70**
FR→RU	41.02/45.01	36.73/38.23	36.26/37.40	37.20/38.54	43.35/44.75	**47.13/48.06**
RU→IT	41.49/46.57	43.84/46.78	39.50/43.74	43.79/45.89	46.52/49.66	**48.66/51.96**
IT→RU	40.57/44.13	38.35/38.71	35.87/38.09	38.40/39.43	45.01/45.48	**47.08/47.49**
TR→FI	21.46/26.46	24.23/28.59	26.14/30.67	24.12/27.90	31.31/32.96	**32.85/34.77**
FI→TR	23.07/28.90	24.86/29.80	23.86/27.54	24.01/28.64	30.48/32.95	**32.74/35.68**
TR→FR	29.13/36.10	32.96/36.58	30.56/34.08	31.31/34.40	38.13/40.63	**41.43/43.88**
FR→TR	27.42/33.52	28.87/31.76	27.42/30.88	26.44/29.13	34.97/37.82	**38.70/42.06**
TR→HR	20.07/24.60	21.99/25.99	22.42/26.68	21.30/25.24	29.34/32.37	**32.43/36.32**
HR→TR	17.41/25.25	24.62/27.35	22.30/26.20	22.09/24.62	29.04/32.61	**34.14/37.09**
TR→IT	28.91/34.56	31.90/34.24	29.66/32.00	29.82/33.44	36.32/38.98	**38.87/42.17**
IT→TR	28.32/34.73	28.11/30.70	27.96/30.39	27.86/29.82	35.09/37.52	**38.19/40.62**
TR→RU	23.59/28.06	24.07/26.20	21.99/26.20	24.55/26.36	31.04/32.00	**33.60/36.16**
RU→TR	24.46/29.18	23.31/27.08	22.58/25.88	25.04/26.35	29.81/32.74	**32.48/35.78**
Avg.	35.78/41.22	36.76/40.06	34.37/38.22	36.22/39.16	41.95/44.54	**45.41/47.88**

Table 6: BLI results with 5k seed translation pairs. BLI prediction accuracy (P@1×100%) is reported in the NN/CSLS format (NN: Nearest Neighbor retrieval without CSLS adjustment; CSLS: CSLS retrieval). Underlined scores denote the highest scores among purely fastText-based methods; **bold** scores denote the highest scores in setups where both fastText and mBERT are allowed.

[1k] **Pairs**	**RCSLS**	**VecMap-Semi**	**LNMap**	**FIPP**	**C1**	**C2 (C1)**
DE→FI	20.97/26.34	23.68/28.33	29.47/32.24	25.56/30.26	37.35/40.85	**40.79/43.77**
FI→DE	21.18/27.01	32.05/35.00	27.64/34.47	31.79/36.73	37.52/40.57	**42.83/44.93**
DE→FR	34.06/41.94	46.17/49.03	43.82/47.21	46.48/50.18	49.82/51.75	**52.11/54.04**
FR→DE	33.89/37.92	42.11/44.34	39.63/42.99	43.30/46.51	46.09/46.82	**48.01/48.16**
DE→HR	19.25/22.59	22.64/27.39	24.26/28.64	21.91/27.18	30.88/35.16	**36.46/40.48**
HR→DE	19.10/23.04	30.98/32.82	25.25/29.46	28.77/31.56	35.35/38.45	**41.19/44.35**
DE→IT	38.81/44.03	46.58/48.72	43.82/47.52	46.01/48.98	48.93/51.28	**50.39/52.53**
IT→DE	36.64/40.83	41.91/44.39	39.69/42.58	42.95/45.94	46.56/47.86	**49.41/49.66**
DE→RU	27.80/32.66	20.97/25.46	27.86/30.73	26.03/30.05	40.11/40.27	**42.15/42.83**
RU→DE	27.82/32.58	36.46/39.08	33.84/37.30	37.98/40.65	42.33/44.21	**45.00/46.99**
DE→TR	14.03/18.21	20.40/23.37	21.39/24.36	18.94/22.85	29.26/32.03	**32.24/34.85**
TR→DE	14.43/18.10	23.22/26.57	20.13/24.55	21.67/25.24	30.83/33.71	**34.45/37.11**
EN→DE	43.00/46.10	46.40/48.20	43.05/45.80	47.95/49.65	49.65/50.40	**51.75/50.85**
DE→EN	43.14/48.25	51.90/54.56	47.16/50.23	50.97/54.41	53.42/56.23	**55.24/57.75**
EN→FI	22.40/28.35	24.30/27.95	29.50/33.60	30.40/34.50	38.60/42.15	**43.75/45.00**
FI→EN	22.70/28.38	37.41/41.15	29.01/35.47	33.68/37.10	39.73/45.51	**42.93/48.77**
EN→FR	49.00/56.50	57.90/60.00	56.85/60.50	59.65/61.60	60.70/61.65	**63.65/62.50**
FR→EN	49.46/55.56	58.35/61.41	54.32/58.41	58.72/61.61	60.48/63.27	**62.65/64.05**
EN→HR	18.65/22.50	21.95/24.95	21.30/25.55	21.70/26.65	32.65/35.65	**39.20/42.35**
HR→EN	16.57/22.88	34.61/37.45	26.35/30.72	29.77/32.93	35.30/40.87	**40.35/47.55**
EN→IT	48.65/55.20	55.15/57.55	54.70/57.60	56.00/58.30	57.70/59.60	**60.70/61.05**
IT→EN	48.22/53.64	56.85/60.78	52.61/56.69	56.59/60.78	59.17/62.64	**61.40/63.67**
EN→RU	31.50/35.50	21.10/25.05	28.50/32.25	32.75/35.15	43.80/42.50	**46.55/46.05**
RU→EN	32.37/36.62	44.37/46.20	36.46/41.17	43.27/46.20	47.25/50.29	**48.35/53.17**
EN→TR	19.35/23.00	24.45/26.70	25.15/27.75	26.40/29.95	36.60/38.15	**39.05/41.05**
TR→EN	19.81/24.65	33.49/37.17	26.94/32.59	29.98/33.76	36.95/42.33	**37.86/43.24**
FI→FR	16.13/22.49	31.84/34.79	25.70/30.01	29.58/33.74	37.05/40.36	**40.67/43.30**
FR→FI	17.69/21.73	21.11/23.95	25.14/28.50	26.49/29.49	34.30/37.61	**37.09/40.56**
FI→HR	15.24/17.24	25.22/29.90	21.86/26.33	23.49/26.90	25.64/30.01	**30.74/34.26**
HR→FI	14.05/18.52	25.04/27.62	23.57/27.83	23.99/27.41	28.67/32.61	**33.46/36.14**
FI→IT	20.13/25.33	32.11/34.68	28.38/31.84	30.27/34.21	35.89/38.99	**40.04/42.88**
IT→FI	19.07/24.60	22.84/26.10	27.80/30.13	27.96/31.01	34.94/37.83	**38.71/41.65**
FI→RU	18.44/21.91	26.69/30.27	23.33/27.69	26.48/30.43	31.42/33.89	**34.73/37.15**
RU→FI	15.72/20.48	29.02/33.11	25.93/31.01	25.93/30.28	32.27/35.31	**34.94/37.35**
HR→FR	17.99/23.04	35.61/39.14	28.35/32.93	30.19/34.67	37.14/41.14	**43.08/45.71**
FR→HR	16.76/20.54	23.80/27.52	24.00/28.45	25.50/28.56	32.70/35.33	**36.26/39.68**
HR→IT	20.52/26.20	36.40/38.77	29.46/33.09	31.93/35.03	37.40/40.24	**42.40/46.19**
IT→HR	18.81/23.72	23.88/28.68	24.81/28.63	26.10/30.44	33.02/35.92	**37.62/41.29**
HR→RU	20.99/24.72	32.40/36.09	29.35/34.30	30.30/34.09	37.30/39.40	**40.72/42.14**
RU→HR	20.32/25.67	34.10/38.08	29.70/33.94	30.91/36.14	34.68/38.92	**38.03/41.17**
IT→FR	55.25/59.95	63.41/65.06	60.93/63.93	63.05/65.22	63.41/65.63	**65.27/66.77**
FR→IT	55.25/59.91	62.13/63.58	60.37/62.80	61.98/64.15	63.11/64.56	**64.46/65.49**
RU→FR	26.72/33.68	42.33/45.42	36.04/40.54	41.91/46.57	46.52/48.87	**48.87/51.28**
FR→RU	27.06/30.83	20.33/24.57	27.57/31.92	29.69/32.90	40.71/40.46	**43.66/43.61**
RU→IT	30.59/35.36	41.91/43.74	38.92/41.80	42.54/44.94	45.10/48.35	**46.46/49.24**
IT→RU	29.82/32.97	22.89/26.10	29.20/31.47	33.49/35.76	41.34/41.50	**43.41/43.57**
TR→FI	13.31/16.03	19.81/24.76	21.73/26.36	21.73/26.20	26.94/29.93	**30.35/32.96**
FI→TR	11.77/15.08	21.97/25.80	19.71/24.17	21.49/25.64	24.96/28.32	**27.80/30.64**
TR→FR	16.67/20.23	30.46/32.85	26.57/31.52	28.27/31.84	35.46/38.82	**38.92/41.59**
FR→TR	14.43/18.37	22.19/25.19	23.02/25.30	21.83/24.37	32.02/35.59	**35.70/38.44**
TR→HR	11.66/13.84	16.19/20.50	19.01/22.15	17.15/21.19	22.74/27.00	**27.85/32.16**
HR→TR	10.10/12.73	19.57/20.67	18.57/21.99	18.36/20.83	22.51/28.25	**28.88/33.04**
TR→IT	17.15/22.31	29.29/31.42	26.94/29.66	26.62/30.56	33.65/36.47	**36.42/39.19**
IT→TR	16.12/20.98	22.22/25.06	23.93/26.10	23.62/26.25	32.66/34.47	**35.50/37.93**
TR→RU	12.94/15.87	13.05/15.55	15.87/19.60	17.04/20.55	25.35/28.12	**29.82/31.95**
RU→TR	11.42/14.77	16.61/18.60	17.02/20.12	20.90/22.89	26.40/29.54	**30.07/33.05**
Avg.	24.73/29.31	32.50/35.56	31.10/34.90	33.00/36.45	38.97/41.74	**42.33/44.77**

Table 7: BLI results with 1k seed translation pairs. BLI prediction accuracy (P@1×100%) is reported in the NN/CSLS format (NN: Nearest Neighbor retrieval without CSLS adjustment; CSLS: CSLS retrieval). Underlined scores denote the highest scores among purely fastText-based methods; **bold** scores denote the highest scores in setups where both fastText and mBERT are allowed.

[1k] **Pairs**	**ET→HU**	**HU→EU**	**EU→ET**
VecMap	35.55	20.03	9.83
FIPP	30.30	11.58	8.22
C1	40.35	20.09	13.00
C2	**44.64**	28.26	21.35
mBERT	15.40	16.97	23.70
mBERT(tuned)	20.59	22.30	28.62
C2(λ=0.4)	-	**34.62**	**36.70**

Table 8: Additional BLI scores on the PanLex-BLI evaluation sets of Vulić et al. (2019); 'mBERT' and 'mBERT (tuned)' refer to using word encodings from mBERT directly for BLI, before and after fine-tuning in Stage C2.

[5k] **Pairs**	**C1 w/o CL**	**C1 w/o SL**	**C1**	**mBERT**	**mBERT(tuned)**	**C1+mBERT**	**C2 (C1)**
DE→∗	35.16/39.30	41.70/45.07	43.43/46.14	8.90/9.39	17.70/18.66	43.13/46.25	**46.24/48.86**
∗→DE	37.24/41.23	43.46/45.85	44.85/46.39	8.86/9.51	18.10/19.21	44.61/46.47	**48.96/50.12**
EN→∗	37.99/41.58	48.41/50.99	49.54/51.31	9.29/9.55	15.08/15.87	49.44/51.55	**53.78/54.31**
∗→EN	46.36/50.16	47.36/51.18	49.21/52.61	10.42/10.71	21.34/22.58	48.96/52.77	**51.22/55.47**
FI→∗	31.92/36.78	33.62/38.21	36.35/39.80	5.73/5.93	12.23/13.23	35.97/40.00	**40.00/43.44**
∗→FI	26.16/31.13	33.07/37.26	35.89/38.82	5.57/5.89	11.99/12.95	35.48/39.05	**39.67/41.97**
FR→∗	38.60/42.41	45.27/48.40	46.81/49.12	9.65/10.18	18.37/19.70	46.65/49.29	**50.29/51.91**
∗→FR	45.30/48.85	47.35/50.82	49.42/51.84	9.86/10.38	20.01/21.10	49.07/51.92	**52.73/54.53**
HR→∗	30.88/35.52	33.95/38.51	36.76/40.22	7.11/7.72	17.52/18.57	36.13/40.40	**41.95/45.53**
∗→HR	26.94/32.19	32.24/36.42	34.67/37.82	7.09/7.54	16.83/17.81	34.23/38.08	**39.13/42.65**
IT→∗	39.06/42.67	45.55/48.39	46.91/48.92	7.47/8.13	18.64/20.18	46.35/48.91	**50.06/51.91**
∗→IT	44.48/47.60	46.35/49.93	48.10/50.99	7.03/7.46	16.24/17.12	47.66/51.07	**51.33/53.85**
RU→∗	37.46/40.84	39.30/42.81	41.77/44.17	1.95/2.29	14.50/15.74	41.56/44.38	**44.25/47.24**
∗→RU	27.85/32.12	39.04/41.46	40.66/42.15	1.38/1.94	11.47/13.25	40.53/42.39	**43.73/45.20**
TR→∗	26.14/30.92	31.12/35.08	34.07/36.61	6.18/6.53	12.10/12.87	33.41/36.81	**36.70/39.86**
∗→TR	22.88/26.74	30.08/34.55	32.83/35.67	6.07/6.28	10.14/10.79	32.09/35.85	**36.52/39.26**
Avg.	34.65/38.75	39.87/43.43	41.95/44.54	7.04/7.46	15.77/16.85	41.58/44.70	**45.41/47.88**

[1k] **Pairs**	**C1 w/o CL**	**C1 w/o SL**	**C1**	**mBERT**	**mBERT(tuned)**	**C1+mBERT**	**C2 (C1)**
DE→∗	33.39/37.54	24.74/32.59	41.40/43.94	8.90/9.39	20.26/20.92	41.46/44.08	**44.20/46.61**
∗→DE	35.21/38.73	24.01/32.08	41.19/43.15	8.86/9.51	20.78/21.10	41.48/43.37	**44.66/46.01**
EN→∗	35.65/39.46	33.21/39.31	45.67/47.16	9.29/9.55	16.92/17.29	46.05/47.56	**49.24/49.84**
∗→EN	44.95/49.02	28.26/39.19	47.47/51.59	10.42/10.71	26.11/26.82	47.08/51.63	**49.83/54.03**
FI→∗	29.34/33.91	13.17/21.10	33.17/36.81	5.73/5.93	15.66/16.13	33.15/36.90	**37.11/40.28**
∗→FI	23.35/28.38	14.12/20.73	33.30/36.61	5.57/5.89	14.80/15.35	33.27/36.83	**37.01/39.63**
FR→∗	36.34/39.49	27.86/34.51	44.20/46.23	9.65/10.18	20.74/21.59	44.15/46.52	**46.83/48.57**
∗→FR	44.06/47.64	28.73/36.32	47.16/49.75	9.86/10.38	23.03/23.59	47.24/49.88	**50.37/52.17**
HR→∗	28.42/33.07	12.40/20.76	33.38/37.28	7.11/7.72	20.41/20.97	33.01/37.38	**38.58/42.16**
∗→HR	24.15/28.84	14.61/20.67	30.33/34.00	7.09/7.54	19.18/19.74	30.49/34.30	**35.17/38.77**
IT→∗	36.71/40.37	29.04/36.45	44.44/46.55	7.47/8.13	22.25/23.29	44.42/46.74	**47.33/49.22**
∗→IT	43.02/46.05	29.42/37.68	45.97/48.50	7.03/7.46	19.27/19.86	45.75/48.54	**48.70/50.94**
RU→∗	35.36/38.69	18.95/27.72	39.22/42.21	1.95/2.29	18.86/19.12	39.09/42.27	**41.67/44.61**
∗→RU	24.33/28.77	20.82/26.62	37.15/38.02	1.38/1.94	14.57/15.74	37.41/38.37	**40.15/41.04**
TR→∗	24.06/28.62	11.39/18.22	30.27/33.77	6.18/6.53	14.80/15.28	30.07/33.92	**33.67/36.89**
∗→TR	20.19/23.73	10.80/17.36	29.20/32.34	6.07/6.28	12.14/12.40	28.69/32.44	**32.75/35.57**
Avg.	32.41/36.39	21.35/28.83	38.97/41.74	7.04/7.46	18.74/19.32	38.93/41.92	**42.33/44.77**

Table 9: Full ablation study on 8 languages, 28 language pairs in both directions with training dictionary sizes of 5k and 1k respectively, that is, 112 BLI setups for each method. $L \to\ast$ and $\ast \to L$ denote the average BLI scores of BLI setups where L is the source and the target language, respectively. BLI prediction accuracy (P@$1\times100\%$) is reported in the NN/CSLS format (NN: Nearest Neighbor retrieval without CSLS adjustment; CSLS: CSLS retrieval). Underlined scores denote the highest scores among purely fastText-based methods; **bold** scores denote the highest scores in setups where both fastText and mBERT are allowed.