RAPO: An Adaptive Ranking Paradigm for Bilingual Lexicon Induction

Zhoujin Tian[*], **Chaozhuo Li**[†], **Shuo Ren, Zhiqiang Zuo, Zengxuan Wen, Xinyue Hu**
Xiao Han, Haizhen Huang, Denvy Deng, Qi Zhang, Xing Xie
Microsoft
`deritt7@gmail.com`, `{cli,renshuo,zhiqzuo,zewen,xinyuehu}@microsoft.com`
`{xiaoha,hhuang,dedeng,qizhang,xingx}@microsoft.com`

Abstract

Bilingual lexicon induction induces the word translations by aligning independently trained word embeddings in two languages. Existing approaches generally focus on minimizing the distances between words in the aligned pairs, while suffering from low discriminative capability to distinguish the relative orders between positive and negative candidates. In addition, the mapping function is globally shared by all words, whose performance might be hindered by the deviations in the distributions of different languages. In this work, we propose a novel ranking-oriented induction model RAPO to learn personalized mapping function for each word. RAPO is capable of enjoying the merits from the unique characteristics of a single word and the cross-language isomorphism simultaneously. Extensive experimental results on public datasets including both rich-resource and low-resource languages demonstrate the superiority of our proposal. Our code is publicly available in `https://github.com/Jlfj345wf/RAPO`.

1 Introduction

Bilingual lexicon induction (BLI) aims at inducing the word translations across two languages based on the monolingual corpora, which is capable of transferring valuable semantic knowledge between different languages, spawning a myriad of NLP tasks such as machine translation (Lample et al., 2018; Artetxe et al., 2018c), semantic parsing (Xiao and Guo, 2014), and document classification (Klementiev et al., 2012). The nucleus of BLI is learning a desirable mapping function to align two sets of independently trained monolingual word embeddings (Mikolov et al., 2013; Ruder et al., 2019; Glavaš et al., 2019). Mikolov et al. (2013) empirically observe that the linear projections are superior to their non-linear counterparts due to the isomorphism across different embedding spaces. Subsequent improvements are successively proposed to advance BLI task by imposing orthogonal constraints (Xing et al., 2015; Conneau et al., 2018), normalizing the embeddings (Artetxe et al., 2016, 2018a; Zhang et al., 2019), reducing the noises (Artetxe et al., 2018b; Yehezkel Lubin et al., 2019), relaxing the hypothesis of isomorphism (Søgaard et al., 2018; Patra et al., 2019), and iteratively refining the seed dictionary (Zhao et al., 2020).

Existing methods (Artetxe et al., 2016, 2018a; Jawanpuria et al., 2020) usually aim at minimizing the distance between the word from the source language and its aligned word in the target language (e.g., *crow* and *cuervo* in Figure 1). However, BLI is essentially a ranking-oriented task because for each source word, we expect to select its top k high-confidence target candidates. Namely, a desirable BLI model should also be capable of distinguishing the relative orders between the positive and negative candidates (e.g., *crow* and *curevo* should be distributed closer than *crow* and *pájaro*). The objective functions used by previous works solely focus on the distances between positive pairs and cannot explicitly provide such important ranking signals, leading to the low discriminative capability.

In addition, conventional BLI models (Mikolov et al., 2013; Xing et al., 2015; Zhao et al., 2020) induce the bilingual space via a shared mapping function, in which different words in the same language tend to be rotated in the same directions. However, several studies (Søgaard et al., 2018; Patra et al., 2019) have demonstrated that the isomorphic assumption may not strictly hold true in general, and thus a global-shared mapping is not the optimal solution. As shown in Figure 1, even for two close languages like English and Spanish, due to deviations in the distributions of different training corpora and insufficient training of low-frequency word embeddings, the optimal mapping directions are slightly shifted for different words. Therefore, the BLI performance could be further

[*] Work is done during internship at Microsoft.
[†] Corresponding author and equal contribution.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 8870–8883
December 7-11, 2022 ©2022 Association for Computational Linguistics

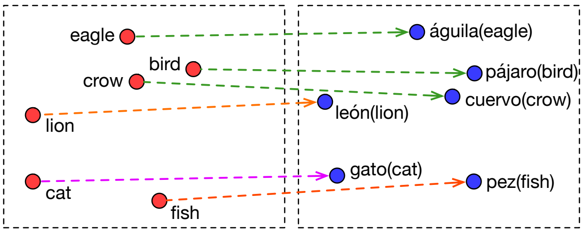


Figure 1: Representation spaces of several aligned words in English (left) and Spanish (right).

improved if we could learn unique or personalized mapping functions for different words. Glavaš and Vulić (2020) first propose to achieve the personalized mappings. However, Glavaš and Vulić (2020) is a non-parametric model, in which the personalized mappings are unlearnable and built upon the heuristic assumptions, which might be unreliable and suffer from low generality.

To address the mentioned limitations under a unified framework, we propose a novel **R**anking-based model with **A**daptive **P**ersonalized **O**ffsets, dubbed **RAPO**. Different from previous works solely relying on the aligned pairs, RAPO is formulated as a ranking paradigm with powerful discriminative capability by incorporating abundant unaligned negative samples. An effective dynamic negative sampling strategy is further proposed to optimize the ranking objectives. Moreover, we integrate a novel personalized adapter into RAPO to learn unique mapping directions for different words. A straightforward strategy is to directly learn an independent mapping matrix for each word, which is resource-consuming and ignores the global isomorphism information. Differently, our personalized adapter learns the unique offset for each word to calibrate the vanilla embedding, and then a shared mapping function is employed to induct lexicons. By organically integrating personalized offsets with shared mapping functions, RAPO enjoys the merits from the unique traits of each word and the global consistency across languages. We further propose a novel Householder projection as the mapping function on the basis of Householder matrices (Householder, 1958), which strictly ensures the orthogonality during the model optimization. We conduct extensive experiments over multiple language pairs in the public MUSE benchmarks (Conneau et al., 2018), including rich- and low-resource languages. Experimental results demonstrate that our proposal consistently achieves desirable performance in both supervised and semi-supervised learning settings.

Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to propose a ranking-based bilingual lexicon induction model RAPO with powerful discriminative capacity.
- We propose a novel personalized adapter to achieve unique mapping direction for each word by adaptively learning the personalized embedding offsets.
- We conduct extensive experiments over popular benchmarks and the results demonstrate the superiority of our proposal.

2 Preliminary

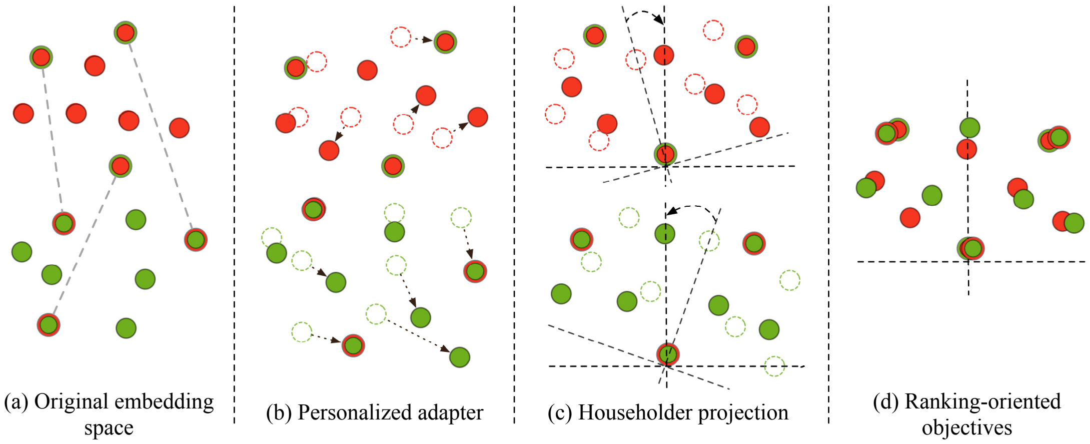
Let $\mathbf{X} \in \mathbb{R}^{d \times n_x}$ and $\mathbf{Y} \in \mathbb{R}^{d \times n_y}$ be monolingual embedding matrices consisting of n_x and n_y words for the source and target languages respectively, and d stands for the embedding size. $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_l, \mathbf{y}_l)\}$ denotes the available aligned seeds, which can be also formulated as two matrices $\mathbf{X}_D$ and $\mathbf{Y}_D \in \mathbb{R}^{d \times l}$. BLI aims to map the source and target words from their original embedding spaces into a shared latent space, in which the mapped source word $\phi_s(\mathbf{x}_i)$ should be close to its matched target word $\phi_t(\mathbf{y}_i)$. ϕ_s and ϕ_t denote the mapping functions for source language and target language, respectively. For the sake of clarification, notations used in this paper are listed in Appendix A.

A widely adopted solution is to set the source mapping function ϕ_s as a linear transformation matrix $\mathbf{W} \in \mathbb{R}^{d \times d}$ and the target mapping function ϕ_t to an identity matrix $\mathbf{I} \in \mathbb{R}^{d \times d}$. The objective function is defined as follows:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \|\mathbf{W}\mathbf{X}_D - \mathbf{I}\mathbf{Y}_D\|_F. \qquad (1)$$

In the inference phase, the distances between the mapped source words and the target ones are utilized as the metrics to select the top-k nearest neighbors as the translations.

As discussed in the section of introduction, this popular induction paradigm suffers from two limitations. First, Formula (1) only focuses on minimizing the distance between the aligned words (e.g., $\mathbf{x}_i$ and $\mathbf{y}_i$). However, BLI task is essentially a ranking problem, which means that the learned mapping function should also be able to distinguish the aligned pair $\{\mathbf{x}_i, \mathbf{y}_i\}$ and the defective one $\{\mathbf{x}_i, \mathbf{y}_j\}$. Second, the globally shared mapping matrix



(a) Original embedding space

(b) Personalized adapter

(c) Householder projection

(d) Ranking-oriented objectives

Figure 2: Overview of the proposed RAPO model. First, the personalized adapter calibrates vanilla embeddings based on the learned personalized offsets. Then, two adjusted embedding spaces are mapped into a shared latent space via the orthogonal Householder projections. Finally, we design the hybrid ranking-oriented objective functions to optimize the model parameters.

W might be inappropriate since the optimal mapping directions of different words tend to be various. Our proposed RAPO is capable of addressing the mentioned challenges under a unified learning framework. We will introduce the details of RAPO in the next section.

3 Methodology

As shown in Figure 2, RAPO consists of three major components. Given the monolingual embeddings and training seeds, the personalized adapter first generates the adaptive offset for each word by exploiting the contextual semantic information. The vanilla embedding spaces are properly calibrated to fit the induction task. After that, we map the adapted word embeddings to a shared latent space via the novel Householder projections, which is capable of ensuring the strict orthogonality and better preserving the isomorphism. Finally, RAPO designs the ranking objectives to distinguish the aligned pairs from the unmatched ones. RAPO can be easily adapted to the supervised and semi-supervised settings, demonstrating its flexibility.

3.1 Personalized adapter

Due to the deviations in the distribution of different corpora and the unbalanced training of word embeddings, recent works demonstrated that the vanilla word embedding space may not be fully trustworthy and proper adjustments contribute to improving induction performance. Previous work (Glavaš and Vulić, 2020) proposes to modify the mapped embedding based on its nearest neighbors in the training dictionary, which is a non-parametric model and might be unreliable. Here we design a novel learnable personalized adaptor, which can be trained through the gradient descent and learn task-relevant personalized offsets.

Given a source word embedding $\mathbf{x}$, adapter first obtains its contextual semantic vector $\bar{\mathbf{x}}$ by averaging the embeddings of its neighbor words $\mathcal{M}_s(\mathbf{x})$ in the source space. Our motivation lies in that a single word can only provide limited information, while a set clustering similar words can assemble the mutual word relationships and provide richer and more accurate information. The contextual semantic vector $\bar{\mathbf{x}}$ is formally defined as:

$$\begin{aligned} \bar{\mathbf{x}} &= \frac{1}{m_s} \sum_{\mathbf{x}_j \in \mathcal{M}_s(\mathbf{x})} \mathbf{x}_j \\ \mathcal{M}_s(\mathbf{x}) &= \{\mathbf{x}_j \mid \langle \mathbf{x}_j, \mathbf{x} \rangle > \tau_s\} \end{aligned} \tag{2}$$

where m_s is the size of $\mathcal{M}_s(\mathbf{x})$ and $\langle , \rangle$ denotes the dot product. τ_s is a hyper-parameter denoting the similarity threshold. Compared to the original word embedding $\mathbf{x}$, the contextual vector $\bar{\mathbf{x}}$ is more informative by incorporating richer semantics.

After that, personalized adapter learns the unique offset for each word based on the contextual semantic vector, which can be effectively optimized by the training objectives. Previous work (Ren et al., 2020) has observed that semantic similar words enjoy stronger isomorphism structures across different embedding spaces. Thus, our motivation lies in that words with similar contextual semantics also tend to have similar personalized offsets.

Specifically, the adapter is implemented as a feed-forward network with a single layer:

$$\mathrm{A}_s(\mathbf{x}) = \sigma(\mathbf{W}_s\bar{\mathbf{x}}) \tag{3}$$

where σ denotes the activation function and it could be linear or non-linear. $\mathbf{W}_s \in \mathbb{R}^{d\times d}$ stands for the learnable parameters. The generated offset vector indicates the personalized offset direction, and is further combined with the vanilla embedding $\mathbf{x}$:

$$\tilde{\mathbf{x}} = \mathbf{x} + \mathrm{A}_s(\mathbf{x}) = \mathbf{x} + \sigma(\mathbf{W}_s\bar{\mathbf{x}}). \tag{4}$$

$\tilde{\mathbf{x}}$ denotes the calibrated word embedding, and will be normalized to the unit length to ensure the consistent value range.

Similarly, for the target word embedding $\mathbf{y}$, the calibrated embedding $\tilde{\mathbf{y}}$ can be calculated as:

$$\tilde{\mathbf{y}} = \mathbf{y} + \mathrm{A}_t(\mathbf{y}) = \mathbf{y} + \sigma(\mathbf{W}_t\bar{\mathbf{y}}) \tag{5}$$

where $\mathrm{A}_t(\mathbf{y})$ is the personalized adapter for target language with learnable parameters $\mathbf{W}_t \in \mathbb{R}^{d\times d}$ and the contextual semantic vector $\bar{\mathbf{y}}$ is obtained in the similar manner.

In a nutshell, the proposed adapter has the following obvious advantages. 1) **Personalization**: the offset is learned based on the contextual semantic features, which are different for various words. 2) **Flexibility**: σ could be either linear or non-linear function to handle different types of language pairs such as close languages (e.g., English-Spanish) or distant ones (e.g., English-Chinese). 3) **Task-relevant**: vanilla word embeddings are unsupervised learned and might be incompatible with the BLI task. The proposed adapter is capable of properly adjusting the original embeddings according to the downstream induction tasks.

3.2 Householder projection

Based on the calibrated embeddings, we further need to design desirable mapping functions to map them into a shared latent space. Previous works (Xing et al., 2015; Conneau et al., 2018; Patra et al., 2019) have demonstrated that the orthogonality of the mapping function is crucial to the model performance. A general approach is to add an extra constraint in the objective function to force the mapping matrix to be orthogonal (i.e., $\min_W ||\mathbf{W}\mathbf{W}^\top - \mathbf{I}||_2$). Nevertheless, such constraints can only achieve an approximate orthogonal matrix instead of the strict one, which may hinder the capability of BLI models in capturing the unsupervised isomorphism information. Here we propose to construct a strict orthogonal mapping function based on the Householder matrices (Householder, 1958; Li et al., 2022), dubbed Householder projection.

Householder matrix represents the reflection about a hyperplane containing the origin. Given a unit vector $\mathbf{v} \in \mathbb{R}^d$, the $d \times d$ Householder matrix $\mathbf{H}$, taking $\mathbf{v}$ as parameter, is defined as $\mathrm{H}(\mathbf{v})$:

$$\mathrm{H}(\mathbf{v}) = \mathbf{I} - 2\mathbf{v}\mathbf{v}^\top \tag{6}$$

where $||\mathbf{v}||_2 = 1$ and $\mathbf{I}$ is the $d \times d$ identity matrix. Given an input vector $\mathbf{z}$, the Householder matrix transforms $\mathbf{z}$ to $\hat{\mathbf{z}}$ by a reflection about the hyperplane orthogonal to the normal vector $\mathbf{v}$:

$$\hat{\mathbf{z}} = \mathrm{H}(\mathbf{v})\mathbf{z} = \mathbf{z} - 2\langle \mathbf{z}, \mathbf{v}\rangle \mathbf{v}. \tag{7}$$

Based on the Householder matrix, we can design a novel Householder projection as the mapping function to ensure strict orthogonal transformations. Householder projection is composed of a set of consecutive Householder matrices. Specifically, given a series of unit vectors $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^n$ where $\mathbf{v}_i \in \mathbb{R}^d$ and n is a positive integer, we define the Householder Projection (HP) as follows:

$$\mathrm{HP}(\mathcal{V}) = \prod_{i=1}^{n} \mathrm{H}(\mathbf{v}_i). \tag{8}$$

We can theoretically prove the following theorem:

Theorem 1. *The image of* HP *is the set of all $n \times n$ orthogonal matrices, i.e.,* $\mathrm{Image}(\mathrm{HP}) = \mathbf{O}(n)$*, $\mathbf{O}(n)$ is the n-dimensional orthogonal group. (See proof in Appendix B)*

Next we will introduce how to employ the Householder projections in the RAPO model. Each language is associated with its unique Householder projection to map words into the shared latent space. Take the source language as an example. Given the calibrated source word embedding $\tilde{\mathbf{x}}$, we employ the source Householder projection with $\mathcal{V}_s = \{\mathbf{v}_1, .., \mathbf{v}_n\}$ as the mapping function:

$$\hat{\mathbf{x}} = \mathrm{HP}(\mathcal{V}_s)\tilde{\mathbf{x}} = \prod_{i=1}^{n} \mathrm{H}(\mathbf{v}_i)\tilde{\mathbf{x}} \tag{9}$$

where n is set to d to fully cover the set of all $d \times d$ orthogonal matrices. Similarly, we employ n unit vectors $\mathcal{U}_t = \{\mathbf{u}_1, .., \mathbf{u}_n\}$ to parameterize the Householder projection for the target language.

Based on Theorem 1, the Householder projection is capable of maintaining strict orthogonality during model optimization such as stochastic gradient descent (SGD), which is theoretically appealing compared to its counterparts. In addition, the efficiency of Householder projection is also guaranteed. The number of learnable parameters in Householder projection (i.e., the set $\mathcal{V}_s$) is $d \times d$, which is identical to the conventional mapping matrix $\mathbf{W}$ in Formula (1). Moreover, the matrix-vector multiplications in Formula (9) can be replaced by vector multiplications following Formula (7), which reduce the time complexity from $O(nd^2)$ to $O(nd)$.

3.3 Ranking-oriented objective function

BLI is a ranking-oriented task as we expect to select the top k target words with high confidence for each source word. However, the training objective functions (e.g., Formula (1)) of previous works (Artetxe et al., 2016, 2018a; Jawanpuria et al., 2020) essentially minimize the distances between the aligned words, which cannot provide sufficient discriminative capacity to rank the candidates. Differently, we propose to optimize the RAPO model by a ranking loss, which empowers our proposal the ability of learning the relative orders between the aligned words and unmatched ones.

Specifically, we adopt a popular pair-wise ranking loss, Bayesian personalized ranking (BPR) (Rendle et al., 2009; Zhang et al., 2022), as the the major training objective function. Given an aligned positive pair $(\mathbf{x}, \mathbf{y})$, the ranking loss is defined as follows:

$$\begin{aligned}\mathcal{L}_r(\mathbf{x}, \mathbf{y}) = \\ -\frac{1}{K} \sum_{\hat{\mathbf{y}}^- \in \mathcal{N}^-} \log \delta(g(\hat{\mathbf{x}}, \hat{\mathbf{y}}) - g(\hat{\mathbf{x}}, \hat{\mathbf{y}}^-))\end{aligned} \tag{10}$$

in which δ is the sigmoid function, and g(,) measures the similarity between two word embeddings. In order to counteract the challenge of hubness problem, we adopt the cross-domain similarity local scaling (CSLS) (Joulin et al., 2018) as the similarity metric, which penalizes the similarity values in dense areas of the embedding distribution.

Set $\mathcal{N}^-$ contains K negative samples, which provides crucial ranking signals for RAPO. Most of the negative sampling strategies can be divided into hard negative sampling and random negative sampling. Following the theoretical analysis of these two strategies (Zhan et al., 2021), we propose to utilize dynamic hard negatives to optimize the top-ranking performance and random negatives to stabilize the training procedure.

Hard negative sampling emphasizes the top-ranking performance and disregards the lower-ranked pairs that hardly affect the lexicon inductions. We employ the dynamic hard negative sampling strategy as the traditional static sampling methods have potential risks in decreasing ranking performance (Zhan et al., 2021). At the beginning of each training epoch, we retrieve the top-K_h candidates for each source word based on the current model parameters, from which the unmatched samples are selected as the hard negative samples. To stabilize the training process, we additionally introduce random negative samples. For each source word in training dictionary, K_r random negative samples are uniformly sampled from the entire target vocabulary. Finally, we can achieve $K = K_h + K_r$ negative samples by merging the hard and random samples.

Additionally, another loss is incorporated to emphasize on the supervised signals by minimizing the Euclidean distance between the aligned words:

$$\mathcal{L}_m(\mathbf{x}, \mathbf{y}) = ||\hat{\mathbf{x}} - \hat{\mathbf{y}}||_2 \tag{11}$$

The final loss of RAPO is the combination of these two objective functions:

$$\begin{aligned}\mathcal{L}(\mathcal{D}, \theta) =& \frac{1}{l} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}} (\mathcal{L}_r(\mathbf{x}, \mathbf{y}) + \lambda_1 \mathcal{L}_m(\mathbf{x}, \mathbf{y})) \\ &+ \lambda_2 ||\theta||_2\end{aligned} \tag{12}$$

where θ denotes the model parameter set, λ_1, λ_2 are hyper-parameters to control the importance of losses and the last item is the L2 regularization to control the range of parameter values.

3.4 Training paradigm

The proposed RAPO model can be employed in both supervised and semi-supervised scenarios. Here we introduce the semi-supervised training procedure of RAPO in Algorithm 1, which iteratively expands the seed dictionary and refines the mapping function. We first calculate the training loss, and optimize the parameters of the adapters and householder projections as shown in line 3-11. After that, we augment the seed dictionary by selecting the mutual nearest CSLS neighbors after each training iteration in line 13-15. In the inference phase, both source and target words are mapped into the shared latent space, and the nearest CSLS neighbor of each source word are selected as its translation. We also conduct complexity analysis in Appendix C.

Algorithm 1: Training procedure of RAPO

```
Input: Monolingual word embeddings $\mathbf{X}$ and $\mathbf{Y}$, training
    dictionary $\mathcal{D}$, number of iterations $C$, learning rate $\alpha$,
    number of epochs $n\_epochs$.
Pre-process contextual semantic vectors (Formula (2)).
for $c \leftarrow 1$ to $C$ do
    for $e \leftarrow 1$ to $n\_epochs$ do
        Generate the negative sample set $\mathcal{N}^-$ including
            both random and dynamic hard negative samples.
        Achieve personalized offsets with Formula (3).
        Calibrate source and target embeddings following
            Formula (4) and (5), respectively.
        Map the source and target words using the
            Householder projection with Formula (9).
        Calculate the loss $\mathcal{L}$ with Formula (12).
        $\theta \leftarrow$ parameters of $\{\mathbf{W}_s, \mathbf{W}_t, \mathcal{V}_s, \mathcal{U}_t\}$
        $\theta \leftarrow \theta - \alpha\partial\mathcal{L}(\mathcal{D}, \theta)$
    end for
    /* dictionary augmentation */
    Calculate mapped embeddings: $\hat{\mathbf{X}}$, $\hat{\mathbf{Y}}$.
    Induce new seeds $\mathcal{D}'$ based on the CSLS similarities
        between $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$.
    $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}'$
end for
```

4 Experiment

4.1 Experimental Settings

Dataset RAPO model is evaluated on the the widely used MUSE dataset (Conneau et al., 2018). Following (Mohiuddin et al., 2020), we select five high-resource language pairs and five low-resource pairs to thoroughly test the model performance. Precision@1 is selected as the measurement (Conneau et al., 2018). We use the data splits in the original MUSE dataset. Detailed statistics could be found in Appendix D.

Baselines We compare RAPO with popular SOTA BLI baselines, including unsupervised methods (Conneau et al., 2018; Artetxe et al., 2018b; Mohiuddin and Joty, 2019; Ren et al., 2020) and semi-supervised/supervised methods (Artetxe et al., 2018a; Conneau et al., 2018; Joulin et al., 2018; Jawanpuria et al., 2019; Patra et al., 2019; Mohiuddin et al., 2020; Zhao et al., 2020). Please refer to Appendix E for the details. For each baseline, we directly report the results in the original papers and conduct experiments with the publicly available code if necessary.

Implementation details Following previous works, vocabularies of each language are trimmed to the 200K most frequent words. The original word embeddings are normalized to enhance performance (Artetxe et al., 2018b), including a length normalization, center normalization and another length normalization. The number of training iterations is set to 5, and the number of training epochs is set to 150 with early stopping. We use the CSLS as the induction metric, and the number of nearest neighbors in the CSLS is set to 10. Adam optimizer is selected to minimize training loss. We only consider 15,000 most frequent words in the seed dictionary augmentation (Zhao et al., 2020). The search spaces of hyper-parameters are presented in Appendix F.

4.2 Main Results

The proposed RAPO model is extensively evaluated over five rich-resource language pairs (en-es, en-fr, en-it, en-ru and en-zh) and five low-resource pairs (en-fa, en-tr, en-he, en-ar and en-et) in both directions, leading to 20 evaluation sets in total. RAPO model is fully trained 5 times over each dataset and we report the average performance. Table 1 and 2 present the Precision@1 scores of different models.

From the results, one can clearly see that the proposed RAPO model achieves best performance over most datasets, and obtain comparable results on other datasets. From the perspective of average performance, RAPO-sup outperforms the best baseline by 0.7% and 1.3% over the rich- and low-resource datasets, and RAPO-semi beats the best baseline by 0.7% and 1.6% under the semi-supervised setting. Such consistent performance gains demonstrate the superiority and generality of RAPO. Besides, RAPO achieves more significant improvements over the distant language pairs (e.g., en-ru, en-fa and en-tr), which indicates that the personalized adapter is capable of bridging the gaps between language pairs by calibrating the vanilla embeddings to fit the BLI task. Overall, such advanced performance of RAPO owes to the appropriate ranking objectives, personalized adaptions and orthogonal projections.

4.3 Ablation Study

To investigate the effectiveness of various components in RAPO, we conduct extensive ablation studies on four datasets, including en-it, en-ru, en-tr and en-he. To avoid the influence of the potential noises introduced by the self-learning, ablation studies are investigated under the supervised setting.

Personalized adaptor Here we study the impact of the personalized adaptor (PA) module. Table 3 reports the experimental results. After remov-

Method	en-es		en-fr		en-it		en-ru		en-zh		avg
	→	←	→	←	→	←	→	←	→	←	
Unsupervised											
(Conneau et al., 2018)	81.7	83.3	82.3	81.1	77.4	76.1	44.0	59.1	32.5	31.4	64.9
(Artetxe et al., 2018b)	82.3	84.7	82.3	83.6	78.8	79.5	49.2	65.6	-	-	-
(Mohiuddin and Joty, 2019)	82.7	84.7	-	-	79.0	79.6	46.9	64.7	-	-	-
(Ren et al., 2020)	82.9	85.3	82.9	83.9	79.1	79.9	49.7	64.7	38.9	35.9	68.3
Supervised											
(Artetxe et al., 2018a)	81.9	83.4	82.1	82.4	77.4	77.9	51.7	63.7	32.3	43.4	67.6
(Joulin et al., 2018)	84.1	86.3	83.3	84.1	79.0	80.7	57.9	67.2	45.9	46.4	71.6
(Jawanpuria et al., 2019)	81.9	85.5	82.1	84.2	77.8	80.9	52.8	67.6	49.1	45.3	70.7
RAPO-sup	84.1	86.1	83.5	84.3	79.3	81.9	58.1	68.0	51.7	45.9	72.3
Semi-Supervised											
(Patra et al., 2019)	84.3	86.2	83.9	84.7	79.3	82.4	57.1	67.7	47.3	46.7	72.0
(Mohiuddin et al., 2020)	80.5	82.2	-	-	76.7	78.3	53.5	67.1	-	-	-
(Zhao et al., 2020)$_{CSS}$	84.5	86.9	**85.3**	85.3	**81.2**	82.7	57.3	67.9	48.2	47.1	72.6
(Zhao et al., 2020)$_{PSS}$	83.7	86.5	84.4	85.5	80.4	82.8	56.8	67.4	48.4	47.5	72.3
(Glavaš and Vulić, 2020)	82.4	86.3	84.5	84.9	80.2	81.9	57.0	67.1	47.9	47.2	71.9
RAPO-semi	**84.5**	**87.0**	85.0	**85.7**	80.8	**83.1**	**59.4**	**68.2**	**51.9**	**47.7**	**73.3**

Table 1: Precision@1 for the BLI task on five rich-resource language pairs. Best results are in **bold** and the best supervised results are underlined. The improvements are statistically significant (sign test, p-value < 0.01)

.

Method	en-fa		en-tr		en-he		en-ar		en-et		avg
	→	←	→	←	→	←	→	←	→	←	
Unsupervised											
(Conneau et al., 2018)	33.4	40.7	52.7	63.5	43.8	57.5	33.2	52.8	33.7	51.2	46.2
(Artetxe et al., 2018b)	30.5	-	46.4	-	36.8	53.1	29.3	47.6	19.4	-	-
(Mohiuddin and Joty, 2019)	36.7	44.5	51.3	61.7	44.0	57.1	36.3	52.6	31.8	48.8	46.5
Supervised											
(Artetxe et al., 2018a)	39.0	42.6	52.2	63.7	47.6	58.0	41.2	55.5	37.4	54.0	49.1
(Joulin et al., 2018)	40.5	42.4	53.8	61.7	52.2	57.9	42.2	55.5	40.0	50.2	49.6
(Jawanpuria et al., 2019)	38.0	40.9	48.6	61.9	43.1	56.7	38.1	53.3	33.7	48.7	46.3
RAPO-sup	41.0	43.9	54.2	64.1	53.5	58.5	43.0	55.3	40.7	55.3	50.9
Semi-Supervised											
(Patra et al., 2019)	38.4	39.3	51.8	59.6	51.6	55.2	41.1	53.9	36.3	48.3	47.6
(Mohiuddin et al., 2020)	36.8	43.7	52.5	65.3	52.5	**59.1**	42.2	57.1	41.2	57.5	50.8
(Zhao et al., 2020)$_{CSS}$	41.4	45.8	53.1	63.8	53.0	57.8	44.1	**57.2**	39.2	49.4	50.5
(Zhao et al., 2020)$_{PSS}$	41.8	46.1	54.0	65.4	49.8	57.4	40.2	55.5	40.9	50.8	50.2
(Glavaš and Vulić, 2020)	40.3	45.2	54.3	64.5	47.5	56.6	41.6	56.4	40.1	52.7	50.1
RAPO-semi	**42.4**	**46.3**	**55.7**	**65.8**	**53.5**	58.7	**44.8**	56.5	**42.5**	**58.1**	**52.4**

Table 2: Precision@1 for the BLI task on five low-resource language pairs. Best results are in **bold** and the best supervised results are underlined. The improvements are statistically significant (sign test, p-value < 0.01)

Models	en-it		en-ru		en-tr		en-he	
	→	←	→	←	→	←	→	←
Best	**79.3**	**81.9**	**58.1**	**68.0**	**54.2**	**64.1**	**53.5**	**58.5**
w/o PA	78.9	80.8	57.6	66.9	53.4	63.5	52.8	57.9
linear	**79.3**	**81.9**	57.9	67.4	53.8	63.9	53.0	58.1
tanh	78.9	80.9	57.8	**68.0**	**54.2**	**64.1**	53.2	**58.5**
sigmoid	79.0	81.3	**58.1**	67.6	53.7	63.8	**53.5**	58.4

Table 3: Ablation study on the personalized adaptor.

Models	en-it		en-ru		en-tr		en-he	
	→	←	→	←	→	←	→	←
Best	**79.3**	**81.9**	**58.1**	**68.0**	**54.2**	**64.1**	**53.5**	**58.5**
w/o HA	78.4	81.0	57.2	67.2	52.9	63.2	52.7	57.2
constraint	78.9	81.2	57.6	67.7	53.6	63.7	53.2	57.9

Table 4: Ablation study on the Householder projection.

ing the personalized adaptor, model performance consistently drops over all the datasets, revealing the importance of personalized embedding adaption. Furthermore, we also investigate the influence of activation function σ in Formula (4) and (5). From Table 3, we can observe that the linear activation function works better for the closer language pairs (e.g., en-it), while the non-linear functions are more suit for the distant pairs. This observation is aligned with previous works (Mohiuddin et al., 2020). Thus, the activation function σ should be carefully tuned based on the data characteristics.

Householder projection Here we aim to study the importance of Householder projections (HP), and results are reported in Table 4. One can see that all the performance declines without the House-

Models	en-it		en-ru		en-tr		en-he	
	→	←	→	←	→	←	→	←
Best	**79.3**	**81.9**	**58.1**	**68.0**	**54.2**	**64.1**	**53.5**	**58.5**
w/o $\mathcal{L}_r$	78.3	80.3	56.9	67.0	53.1	63.0	52.2	57.4
w/o $\mathcal{L}_m$	79.0	81.4	57.7	67.7	53.5	63.5	52.9	58.0

Table 5: Ablation study on the objective functions.

holder projections. After employing the conventional orthogonal constraint $\mathbf{W}\mathbf{W}^\top = \mathbf{I}$, the performance is improved but still inferior to the Householder projections. We can get the following two conclusions: 1) the orthogonal mapping functions can boost the BLI performance; 2) the Householder projections are capable of ensuring strict orthogonality and thus obtaining better results.

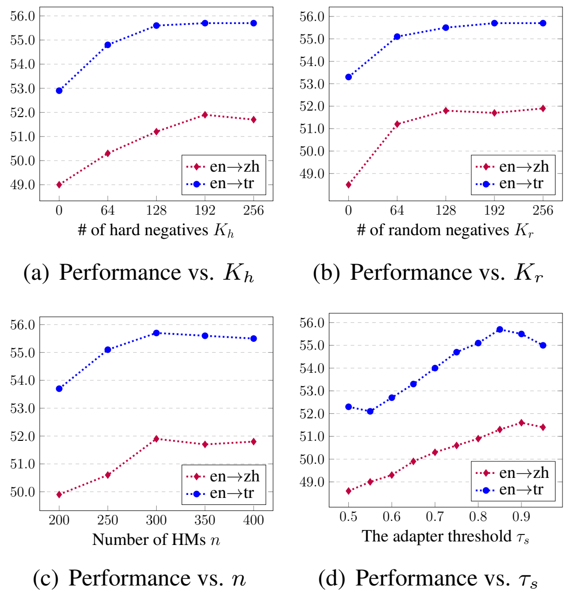
Training objective functions As shown in Formula (12), the loss of RAPO includes two parts: the ranking loss $\mathcal{L}_r$ and the MSE loss $\mathcal{L}_m$. Table 5 presents the model performance without different objectives. We can see that both models present performance decay over all the datasets, which verifies both objective functions would benefit the BLI task. Without the ranking loss, RAPO presents more significant performance drop compared to the MSE loss. It reveals that the BLI task is essentially a ranking problem and thus the ranking loss $\mathcal{L}_r$ would be more important.

4.4 Parameter sensitivity analysis

The parameter sensitivity analysis is conducted on four core hyper-parameters: the number of hard negative samples K_h, the number of random negative samples K_r, the number of Householder matrices (HM) n in Formula (8) and the adaptor threshold τ_s in Formula (2). The sensitivity analysis is conducted on the en→zh and en→tr datasets under the semi-supervised learning scenario.

Number of negative samples Figure 3(a) and Figure 3(b) demonstrate the performance curves of the number of hard negative samples K_h and random samples K_r, respectively. One can clearly see that both types of negative sampling strategies contribute to improving model performance, which indicates that each strategy provides its unique crucial signals to guide the optimization process.

Number of Householder matrices As shown in Figure 3(c), with the increases of n, the performance of RAPO first increases and then keeps steady. This is reasonable as more Householder matrices means the projection could be conducted in a larger rotation space, which benefits the induction performance at the beginning. When n is equal to or larger than the dimension of word embeddings $d = 300$, the Householder projection reaches the peak of its expressivity and thus cannot bringing more benefits.



(a) Performance vs. K_h (b) Performance vs. K_r

(c) Performance vs. n (d) Performance vs. τ_s

Figure 3: Parameter sensitivity analysis.

Threshold of adaptor The hyper-parameter τ_s denotes the threshold to select the neighbor words to generate the contextual semantic vectors. As shown in Figure 3(d), model performance first increases and then significant declines when τ_s is enlarged from 0.5 to 0.95. On one hand, a small τ_s incorporates more neighbor words to provide richer semantics, while also aggravates the risk of introducing noises. On the other hand, a large τ_s ensures the reliability of the contextual vectors but may lead to the scarce neighborhood information. Thus, τ_s should be carefully tuned to find the balance between the contextual semantics and potential noises.

5 Conclusion

In this paper, we propose a novel model RAPO for bilingual lexicon induction. Different from previous works, RAPO is formulated as a ranking paradigm, which is more suitable to the nature of studied tasks. Two novel modules are further employed by deeply mining the unique characteristics of BLI task: the Householder projection to ensure the strict orthogonal mapping function, and the personalized adapter to learn unique embedding offsets for different words. Extensive experimental results on 20 datasets demonstrate the superiority of our proposal.

Limitations

Despite the promising performance of the proposed RAPO model, it may suffer from the following limitations:

- RAPO has more hyper-parameters than the previous works, leading to the exhausting and time-consuming hyper-parameter tuning process.
- Though RAPO has achieved the best performance over almost all the datasets, it fails in few datasets such as en→tr, which might be caused by the insufficient hyper-parameter tuning.
- The supervised signals are indispensable to our proposal, and thus RAPO cannot be applied to the unsupervised learning setting without any labeled data.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5012–5019.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. 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 (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018c. Unsupervised neural machine translation. In *International Conference on Learning Representations*.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305.

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.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

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*, 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*, pages 7548–7555, Online. Association for Computational Linguistics.

Alston S Householder. 1958. Unitary triangularization of a nonsymmetric matrix. *Journal of the ACM (JACM)*, pages 339–342.

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.

Pratik Jawanpuria, Mayank Meghwanshi, and Bamdev Mishra. 2020. Geometry-aware domain adaptation for unsupervised alignment of word embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3052–3058, Online. Association for Computational Linguistics.

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*, pages 2979–2984, Brussels, Belgium. Association for Computational Linguistics.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, pages 1459–1474, Mumbai, India. The COLING 2012 Organizing Committee.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Rui Li, Jianan Zhao, Chaozhuo Li, Di He, Yiqi Wang, Yuming Liu, Hao Sun, Senzhang Wang, Weiwei Deng, Yanming Shen, et al. 2022. House: Knowledge graph embedding with householder parameterization. *ICML 2022*.

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)*, 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, Volume 1 (Long and Short Papers)*, pages 3857–3867, Minneapolis, Minnesota. 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*, pages 184–193, Florence, Italy. 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*, pages 3476–3485, Online. Association for Computational Linguistics.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: bayesian personalized ranking from implicit feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, pages 452–461.

Sebastian Ruder, Ivan Vulic, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *J. Artif. Intell. Res.*, 65:569–631.

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 (Volume 1: Long Papers)*, pages 778–788, Melbourne, Australia. Association for Computational Linguistics.

Min Xiao and Yuhong Guo. 2014. Distributed word representation learning for cross-lingual dependency parsing. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 119–129, Ann Arbor, Michigan. 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*, pages 1006–1011, Denver, Colorado. Association for Computational Linguistics.

Noa Yehezkel Lubin, Jacob Goldberger, and Yoav Goldberg. 2019. Aligning vector-spaces with noisy supervised lexicon. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 460–465, Minneapolis, Minnesota. Association for Computational Linguistics.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 1503–1512.

Mozhi Zhang, Keyulu Xu, Ken-ichi Kawarabayashi, Stefanie Jegelka, and Jordan Boyd-Graber. 2019. Are girls neko or shōjo? cross-lingual alignment of non-isomorphic embeddings with iterative normalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3180–3189, Florence, Italy. Association for Computational Linguistics.

Yiding Zhang, Chaozhuo Li, Xing Xie, Xiao Wang, Chuan Shi, Yuming Liu, Hao Sun, Liangjie Zhang, Weiwei Deng, and Qi Zhang. 2022. Geometric disentangled collaborative filtering. *SIGIR 2022*.

Xu Zhao, Zihao Wang, Hao Wu, and Yong Zhang. 2020. Semi-supervised bilingual lexicon induction with two-way interaction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2973–2984, Online. Association for Computational Linguistics.

A Notations

For the sake of clarification, notations used in this paper are listed in Table 6 .

B Proofs of Theorem 1

Householder projection is composed of a set of consecutive Householder matrices. Specifically, given a series of unit vectors $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^n$ where $\mathbf{v}_i \in \mathbb{R}^d$ and n is a positive integer, we define the Householder Projection (HP) as follows:

$$\mathrm{HP}(\mathcal{V}) = \prod_{i=1}^{n} H(\mathbf{v}_i). \tag{13}$$

The image of HP is the set of all $n \times n$ orthogonal matrices, i.e., $\mathrm{Image(HP)} = \mathbf{O}(n)$, $\mathbf{O}(n)$ is the n-dimensional orthogonal group.

We first prove that the image of HP is a subset of $\mathbf{O}(n)$. Note that each Householder matrix is orthogonal. Therefore, the product of n Householder matrices is also an orthogonal matrix, i.e., $\mathrm{Image(HP)} \subset \mathbf{O}(n)$.

Then we also prove that its converse is also valid, i.e., any $n \times n$ orthogonal matrix Q can be decomposed into the product of n Householder matrices. From the Householder QR decomposition (Householder, 1958), we can upper triangularize any full-rank matrix $W \in \mathbb{R}^{n \times n}$ by using $n - 1$ Householder matrices, i.e.,

$$H(\mathbf{v}_{n-1})H(\mathbf{v}_{n-2}) \cdots H(\mathbf{v}_1)W = R,$$

where $R \in \mathbb{R}^{n \times n}$ is an upper triangular matrix and its first $n - 1$ diagonal elements are all positive.

When Household QR decomposition is performed on an orthogonal matrix Q, we can get:

$$H(\mathbf{v}_{n-1})H(\mathbf{v}_{n-2}) \cdots H(\mathbf{v}_1)Q = R.$$

Note that R here is both upper triangular and orthogonal (i.e., $RR^T = I$) since it is a product of n orthogonal matrices. It establishes that R is a diagonal matrix, where the first $n - 1$ diagonal entries are equal to $+1$ and the last diagonal entry is either $+1$ or -1.

If the last diagonal entry of R is equal to $+1$, i.e., $R = I$, we can set $\mathbf{v}_n = (0, \ldots, 0, 0)^\top \in \mathbb{R}^n$ and consequently get

$$H(\mathbf{v}_{n-1})H(\mathbf{v}_{n-2}) \cdots H(\mathbf{v}_1)Q = I = H(\mathbf{v}_n).$$

As each Householder matrix $H(\mathbf{v}_i)$ is its own inverse, we obtain that

$$Q = H(\mathbf{v}_1) \cdots H(\mathbf{v}_{n-1})H(\mathbf{v}_n).$$

If the last diagonal entry of R is equal to -1, we can set $\mathbf{v}_n = (0, \ldots, 0, 1)^\top \in \mathbb{R}^n$ and get the same conclusion.

From the above we can see that any $n \times n$ orthogonal matrix can be decomposed into the product of n Householder matrices, i.e., $\mathbf{O}(n) \subset \mathrm{Image(HP)}$. All in all, we have $\mathrm{Image(HP)} = \mathbf{O}(n)$.

C Complexity analysis

Parameter complexity analysis The learnable parameters in RAPO come from two modules: personalized offset adapter and Householder projections. The learnable parameters in adapter is matrix W_s with the shape of d^2. The learnable parameters in Householder transformation include n d-dimension unit vectors $\mathcal{V}_s = \{\mathbf{v}_1, .., \mathbf{v}_n\}$, which has a total size of nd. Therefore, the number of parameters for each language is $O(d^2 + nd)$. In our experiments, the number of householder reflection n is the same to d. Thus, the final parameter complexity of RAPO is $O(d^2)$, which is same to the previous models .

Time complexity analysis Given a word embedding $\mathbf{x}$ of dimension d, we first need to find its contextual semantic vector $\bar{\mathbf{x}}$ of dimension d, which can be computed in advance and stored in an efficient lookup table. Next, we calibrate embedding $\tilde{\mathbf{x}}$ using the personalized adapter according to Formula (4), which has a time complexity of $O(d^2)$. Then, the calibrated embedding $\tilde{\mathbf{x}}$ will be mapped to the shared latent space through Householder projection based on the Formula (9). The time complexity of Formula (9) is $O(nd^2)$, in which n matrix-vector multiplications incur high computational costs. However, it is worth noting that these matrix multiplications can be replaced by the vector operations. Formally, based on Equation (7), the j-th matrix-vector multiplication can be expressed as:

$$\begin{aligned} \hat{\mathbf{x}}_j &= \mathrm{H}(\mathbf{v}_j)\hat{\mathbf{x}}_{j-1} \\ &= \hat{\mathbf{x}}_{j-1} - 2\left\langle \hat{\mathbf{x}}_{j-1}, \mathbf{v}_j \right\rangle \mathbf{v}_j \end{aligned} \tag{14}$$

where $\hat{\mathbf{x}}_0 = \tilde{\mathbf{x}}$. Through such iterated vector operations, the time complexity can be reduced to $O(nd)$ or $O(d^2)$. Therefore, the time complexity of mapping function in RAPO is $O(2d^2)$, which is linearly comparable with previous methods.

D Dataset statistics

The widely used MUSE dataset (Conneau et al., 2018) consists of FastText monolingual embed-

Symbol	Shape	Description
$n_x(n_y)$	$\mathbb{R}$	Vocabulary size of source(target) language
d	$\mathbb{R}$	Embedding size
$\mathbf{X}$	$\mathbb{R}^{d\times n_x}$	Monolingual embedding matrices consisting of n_x words for source language
$\mathbf{Y}$	$\mathbb{R}^{d\times n_y}$	Monolingual embedding matrices consisting of n_y words for target language
$\mathcal{D}$	–	Set of available aligned seed dictionary
l	$\mathbb{R}$	Size of $\mathcal{D}$
$\phi_s(\phi_t)$	–	Mapping function for source(target) language
$\bar{x}(\bar{y})$	$\mathbb{R}^d$	Contextual semantic vector of source word $\mathbf{x}$ (target word $\mathbf{y}$)
$\tau_s(\tau_t)$	$\mathbb{R}$	Similarity threshold of personalized adapter for source(target) language
$\mathcal{M}_s(\mathcal{M}_t)$	–	Set of contextual semantic neighbor words in the source(target) space
$m_s(m_t)$	$\mathbb{R}$	Size of $\mathcal{M}_s(\mathcal{M}_t)$
$W_s(W_t)$	$\mathbb{R}^{d\times d}$	Learnable parameters of personalized adapter for source(target) language
σ	–	Activation function used in personalized adapter
$\tilde{x}(\tilde{y})$	$\mathbb{R}^d$	Calibrated embedding of source word $\mathbf{x}$ (target word $\mathbf{y}$)
$\mathcal{V}_s(\mathcal{U}_t)$	–	Set of unit vectors to parameterize the Householder projection for the source(target) language
n	$\mathbb{R}$	Number of unit vectors to parameterize the Householder projection
$\hat{x}(\hat{y})$	$\mathbb{R}^d$	Mapped embedding in the shared latent space of source word embedding $\mathbf{x}$ (target word $\mathbf{y}$)
$\mathcal{N}^-$	–	Set of negative samples
K	$\mathbb{R}$	Number of negative samples
K_h	$\mathbb{R}$	Number of dynamic hard negative samples
K_r	$\mathbb{R}$	Number of random negative samples
θ	–	Parameters of RAPO, including $\{\mathbf{W}_s, \mathbf{W}_t, \mathcal{V}_s, \mathcal{U}_t\}$

Table 6: Notations used in this paper.

Statics	en-es		en-fr		en-it		en-ru		en-zh	
	→	←	→	←	→	←	→	←	→	←
# train set	11977	8667	10872	8270	9657	7364	10887	7452	8728	8891
# test set	2975	2416	2943	2342	2585	2102	2447	2069	2230	2483
# unique source words	6500	6500	6500	6500	6500	6500	6500	6500	6500	6500
# unique target words	12411	8373	11524	8309	10762	7562	11491	6811	8102	8618

Table 7: Statistics of rich-resource language pairs in MUSE.

dings of 300 dimensions (Bojanowski et al., 2017) trained on Wikipedia monolingual corpus and gold dictionaries for many language pairs divided into training and test sets. We conduct extensive experiments over multiple language pairs in the MUSE dataset, including five popular rich-resource language pairs (French (fr), Spanish(es), Italian (it), Russian (ru), Chinese (zh) from and to English(en)) and five low-resource language pairs (Faroese(fa), Turkish(tr), Hebrew(he), Arabic(ar), Estonian(er) from and to English(en)), totally 20 BLI datasets considering bidirectional translation. Table 7 and 8 summarizes the detailed statistics of rich- and low-resource language pairs, respectively.

E Related Work

Recent proposed work on BLI can be mainly divided into two categories. The first is unsupervised learning methods, which induces dictionaries according to the characteristics of the embedding space. The second is the supervised and semi-supervised learning methods, which trains the model based on a small seed dictionary.

Unsupervised method The major challenge of unsupervised methods is to build high-quality initial seed dictionary. Conneau et al. (2018) are the first to show impressive results for unsupervised word translation by pairing adversarial training with effective refinement methods. Given two

Statics	en-fa		en-tr		en-he		en-ar		en-et	
	→	←	→	←	→	←	→	←	→	←
# train set	8869	8510	9771	8793	9634	7737	11571	7534	8261	6509
# test set	2148	2202	2261	2291	2379	2195	2695	2061	1991	1817
# unique source words	6498	6451	6498	6442	6495	6462	6500	6453	6500	6500
# unique target words	8837	8691	9259	8589	10018	7458	11787	6144	8991	6796

Table 8: Statistics of low-resource language pairs in MUSE.

Hyperparameter	Search Space	Type
K_h	$\{64, 128, 192, 256\}$	Choice
K_r	$\{64, 128, 192, 256\}$	Choice
σ	{ none, tanh, sigmoid }	Choice
lr	$[0.001, 0.003]$	Range
τ_s	$[0.7, 0.99]$	Range
τ_t	$[0.7, 0.99]$	Range
λ_1	$[0.5, 2.5]$	Range
λ_2	$[0.001, 0.1]$	Range

Table 9: Hyper-parameter search space.

monolingual word embeddings, they proposed an adversarial training method, where a linear mapper (generator) plays against a discriminator. They also impose the orthogonality constraint on the mapper. After adversarial training, they use the iterative Procrustes solution similar to their supervised approach. Artetxe et al. (2018b) learn an initial dictionary by exploiting the structural similarity of the embeddings in an unsupervised way. They propose a robust self-learning to improve it iteratively.Mohiuddin and Joty (2019) use adversarial autoencoder for unsupervised word translation. They use linear autoencoders in their model, and the mappers are also linear. Ren et al. (2020) propose a graph-based paradigm to induce bilingual lexicons. They first build a graph for each language with its vertices representing different words. Then they extract word cliques from the graphs and align the cliques of two languages to induce the initial word translation solution. Their methods achieved SOTA results in the unsupervised setting.

Supervised and Semi-supervised methods Supervised and semi-supervised methods mainly focus on learning more efficient mapping function to improve the induction performance. Artetxe et al. (2018a) propose a multi-step framework based on their unsupervised method. Their framework consists of several steps: whitening, orthogonal mapping, re-weighting, de-whitening, and dimensionality reduction. Joulin et al. (2018) proposed to directly include the CSLS criterion in the learning object and maximum the CSLS score between the translation pairs of seed dictionary in order to make learning and inference consistent. Jawanpuria et al. (2020) proposed to map the source and target words from their original embedding spaces to a common latent space via language-specific orthogonal transformations. They further define a learnable Mahalanobis similarity metric, which allows for a more effective similarity comparison of embeddings. Søgaard et al. (2018) empirically show that even closely related languages are far from being isomorphic and Patra et al. (2019) propose a semi-supervised technique that relaxes the isomorphic assumption while leveraging both seed dictionary pairs and a larger set of unaligned word embeddings. Mohiuddin et al. (2020) uses non-linear mapping in the latent space of two independently pre-trained autoencoders, which is also independent of the isomorphic assumption. Zhao et al. (2020) design two message-passing mechanisms in semi-supervised setting to transfer knowledge between annotated and non-annotated data, named prior optimal transport and bi-directional lexicon update respectively. Glavaš and Vulić (2020) proposed to move each point along an instance-specific translation vector estimated from the translation vectors of nearest neighbours in training dictionary. Notably, comparing with these supervised and semi-supervised methods, our RAPO model aims at optimizing ranking-base objectives , which is more suitable to the induction task. Furthermore, with proposed personalized adapter and Householder projection, RAPO enjoys the merits from the unique traits of each word and the global consistency across languages, which is capable of improving induction performance on different language pairs consistently.

F Hyper-parameter search

The hyper-parameters are tuned by the random search (Bergstra and Bengio, 2012) for each BLI dataset, including number of dynamic hard negative samples K_h, number of negative samples K_r, activation function used in personalized adapter σ, learning rate lr, similarity threshold of personalized adapter for source and target language τ_s,τ_t, and the weights in loss function λ_1, λ_2. The hyperparameter search space is shown in Table 9.