Low-Rank Adaptation Approach for Vietnamese-Bahnaric Lexical Mapping from Non-Parallel Corpora

La Cam Huy*†, Le Quang Minh*†, Tran Ngoc Oanh*†, Le Duc Dong*†, Duc Q. Nguyen*†, Nguyen Tan Sang*†,
Tran Quan*†, Tho Quan*†‡

*Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
†Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Vietnam
‡Corresponding author: qttho@hcmut.edu.vn

***Abstract*—Bilingual dictionaries are vital tools for automated machine translation. Leveraging advanced machine learning techniques, it is possible to construct bilingual dictionaries by automatically learning lexical mappings from bilingual corpora. However, procuring extensive bilingual corpora for low-resource languages, such as Bahnaric, poses a significant challenge. Recent studies suggest that non-parallel corpora, supplemented with a handful of anchor words, can aid in the learning of these mappings, which contain parameters for automated translation between source and target languages. The prevailing methodology involves using Generative Adversarial Networks (GANs) and solving the Procrustes orthogonal problem to generate this mapping. This approach, while innovative, exhibits instability and demands substantial computational resources, posing potential issues in rural regions where Bahnaric is spoken natively. To mitigate this, we propose a low-rank adaptation strategy, where the limitations of GANs can be circumvented by directly calculating the rigid transformation between the source and target languages. We evaluated our approach using the French-English dataset, and a low-resource dataset, Vietnamese-Bahnaric. Notably, the Vietnamese-Bahnaric lexical mapping produced by our method is valuable not only to the field of computer science, but also contributes significantly to the preservation of Bahnaric cultural heritage within Vietnam's ethnic minority communities.**

***Index Terms*—Low-rank adaptation, lexical mapping, low-resource language, Kabsch algorithm**

I. Introduction

The construction of bilingual dictionaries represents a valuable endeavor for both the computational linguistics and computer science communities. This process necessitates the accumulation, classification, and presentation of word pairs and their corresponding translations in two languages [1]. Historically, this task has entailed the use of reliable linguistic resources, bilingual documents, and consultations with native speakers to ensure precision. However, with recent developments in Artificial Intelligence (AI), it is now feasible to apply machine learning algorithms to train language models capable of comprehending and generating translations between two languages [2]. Such advancements demonstrate the intersection of AI and linguistics, revolutionizing the way we approach bilingual dictionary construction.

However, machine translation methods utilizing machine learning techiques typically rely heavily on a significant volume of parallel bilingual corpora for training, especially in the context of deep learning models [3]. This poses a substantial challenge, particularly for low-resource languages such as Bahnaric, where obtaining such parallel language data is notably difficult. Recent research proposes the construction of a lexical mapping between the source and target languages without the necessity for extensive parallel corpora. This is achieved by learning the mapping between language embedding spaces with the aid of selected anchor words. These anchor words can be automatically extracted or manually designated by linguistic specialists. Figure 1 illustrates the approach at a theoretical level. It begins with two language embedding spaces, one for English and the other for French, each with arbitrary shapes. The mapping process endeavors to convert the embedding space of the source language into that of the target language. Subsequently, adjustments are made to minimize the disparity between the shapes of these two spaces.

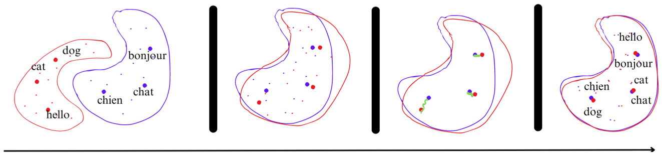

Figure 1: Overview of mapping process.

To isolate the problem of finding the mapping, current state-of-the-art (SOTA) approach [4] presupposes that the two languages under consideration possess analogous structures. Consequently, after training two distinct embedding models, their embedding point cloud shapes are similar [5]. With this assumption, Generative Adversarial Networks (GANs) are then employed to compute the linear mapping matrix $\mathbf{R} \in \mathbb{R}^{n \times n}$. During the refinement phase, this method constructs a synthetic bilingual dictionary containing only high-frequency words, serving as anchors to compute the refined mapping matrix $\mathbf{R}^{'} \in \mathbb{R}^{n \times n}$. However, this method exhibits three primary disadvantages, both theoretically and practically. From a theoretical standpoint, assuming similar embedding point cloud shapes and according to the geometric transformation theories [6], [7], the transformation α between point clouds must operate within the n-dimensional special Euclidean group ($SE(n)$ group) [8], $\alpha \in SE(n)$. Additionally, based on the theory of special Euclidean group,

$$SE(n) = T(n) \rtimes SO(n). \tag{1}$$

Without any enforcement, $\mathbf{R}, \mathbf{R}^{'} \in O(n)$, leading to embedding points of corresponding words in two languages failing to align after transformation (Figure 2). This stems from the group $O(n)$ containing reflection and omitted translation actions within the group $T(n)$. From a practical perspective, constructing bilingual dictionaries with less than 100 words in low-resource languages is conceivable [9], rendering automatic identification of anchor words unnecessary in general use-cases. In certain instances, should the automatically detected anchors deviate from the correct mapping, the resultant computation of the transformation may yield incorrect or erroneous results, as illustrated in Figure 3. Additionally, the adversarial training process in GANs may be unstable [10], resulting in potential model collapse.

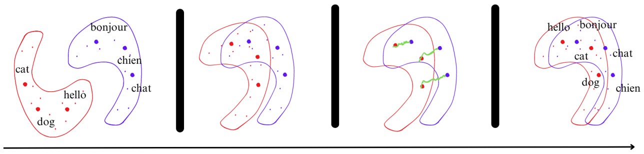

Figure 2: Missing translation problem

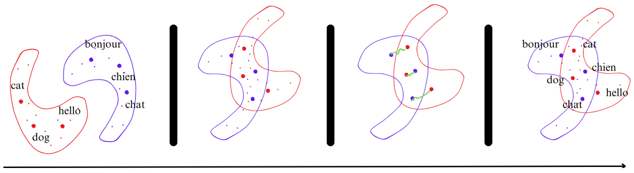

Figure 3: False anchor detection problem

Another challenge associated with low-resource languages is the scarcity of available documents. Without sufficient data, deep learning-based embedding models are not well learned, which may contradict our assumption. To mitigate this, without the need of parallel corpora, data augmentation, via modern techniques, can foster robust embedding models without any further data collection costs [11].

In this study, we propose an effective method known as **A**ugmenting and **S**ampling with **K**absch (ASK) to address the data scarcity in low-resource languages and the aforementioned issues of the SOTA approach. By augmenting the available low-resource language data and utilizing the Kabsch algorithm [12] to fine-tune embedding models with randomly sampled anchor words, we create the transformation $\alpha \in SE(n)$ to map the source embedding space to the target one. Our contributions are outlined as follows.

- Implementation of contemporary data augmentation techniques, including sentence boundary augmentation and multitask learning data augmentation, to enhance low-resource language data, thus improving the performance of embedding model.
- Adaptation of the Kabsch algorithm with randomly sampled anchors to fine-tune and compute the mapping of two language embedding spaces.
- Execution of experiments to assess the efficacy of our proposed method across various settings, including the well-known French-English dictionary and the low-resource Vietnamese-Bahnaric dictionary, underlines the importance of data augmentation and demonstrates the correctness and efficiency of our approach.

II. Related Works

A. Similarity between embedding spaces across languages

Recent advancements in the field of language representation have unveiled compelling insights into the structural similarities that exist across various languages. A study by [13], [14], [15] reveals that languages sharing a similar grammatical structure tend to exhibit corresponding shapes within their embedding point clouds when analyzed using identical embedding models. This congruence between different language spaces is not merely coincidental but is likely indicative of underlying linguistic parallels that manifest in the syntactic and semantic dimensions of the languages. The discovery has profound implications for cross-lingual modeling and machine translation, as it could lead to more efficient algorithms for mapping between different language spaces [15]. However, the correctness of an embedding model strongly depends on the training dataset. In case the two languages have analogous structures, if one of them does note have richdicuous dataset, their embedding point clouds could be significant different.

B. Lexical mapping for low-resource languages

Lexical mapping, the computational process of aligning words or phrases across different languages, represents an active area of research with critical implications for the creation of bilingual dictionaries, especially for low-resource languages such as those spoken by ethnic minority groups. This research is essential for the enhancement of machine translation systems that rely on these dictionaries. Lexical mapping solutions can be broadly divided into three categories: (i) methods requiring parallel data; (ii) methods necessitating only a few parallel anchors; and (iii) methods operating with non-parallel data.

Approaches utilizing parallel data typically exhibit superior performance, with techniques ranging from the normalization and application of orthogonal mapping for translation [16] to the development of extensive multilingual word embeddings [17]. However, obtaining sufficient parallel data for low-resource languages remains a significant challenge, limiting the effective deployment of deep learning-based methods in practical applications.

In response to this limitation, research has explored solutions that do not require parallel data. A recent example involves the utilization of adversarial training to automatically identify anchor words, which are then used to compute transformations between embedding spaces [4]. Though this approach circumvents the need for parallel corpora and achieves SOTA performance among non-parallel data approaches, its performance remains markedly below that of methods relying on parallel corpora.

It is worth noting that the construction of a small bilingual dictionary is often feasible, making methods that use such dictionaries as anchors particularly promising. These approaches are designed to strike a balance between data requirements and methodological performance, addressing a critical trade-off in the quest to automate the process of bilingual dictionary creation and enhance machine translation capabilities.

C. Rigid transformation and Special Euclidean Group

A rigid transformation, also known as a Euclidean transformation or isometry [18], is a geometric transformation that preserves distance between every pair of points. In more formal terms, a transformation α is considered rigid if for any two points A and B, the distance between A and B is the same as the distance between $\alpha(A)$ and $\alpha(B)$. The Euclidean group [19], denoted as $E(n)$, is the group of all Euclidean transformations in n-dimensional Euclidean space. It is a mathematical structure that encodes the geometry of Euclidean space and captures the ways objects can be moved around without changing their shape or size. Transformations in $E(n)$ group can be decomposed into components in two subgroups which are rotation ($O(n)$) and translation ($T(n)$) groups (Equation 2).

$$E(n) = T(n) \rtimes O(n) \tag{2}$$

In linear algebra, transformation in $E(n)$ can be also defined as Equation 3.

$$\begin{aligned} E(n) = \Big\{ \mathbf{A} \Big| \mathbf{A} = \left[\begin{array}{c:c} \mathbf{R} & \mathbf{t} \\ \hdashline 0_{1\times n} & 1 \end{array}\right], \mathbf{R} \in \mathbb{R}^{n\times n}, \\ \mathbf{t} \in \mathbb{R}^{n}, \mathbf{R}^\top \mathbf{R} = \mathbf{R}\mathbf{R}^\top = \mathbf{I} \Big\} \end{aligned} \tag{3}$$

Assuming that x is a point in a n-dimensional Euclidean space, the transformation α can be expressed as

$$\alpha(x) = \mathbf{R}x + \mathbf{t}. \tag{4}$$

However, in ($n > 2$)-dimensional spaces, the transformation can include reflections, which is unnecessary in some usecases such as moving aerospace rocket in spaces. Therefore, theoretically, we do have a subgroup known as special Euclidean group ($SE(n)$) which includes only the isometries that preserve orientation. This means it consists of translations and rotations, but excludes reflections. The term “special"in the name refers to the preservation of orientation. Formal definition of $SE(n)$ in linear algebre is illustrated in (5).

$$\begin{aligned} SE(n) = \Big\{ \mathbf{A} \Big| \mathbf{A} = \left[\begin{array}{c:c} \mathbf{R} & \mathbf{t} \\ \hdashline 0_{1\times n} & 1 \end{array}\right], \mathbf{R} \in \mathbb{R}^{n\times n}, \\ \mathbf{t} \in \mathbb{R}^{n}, \mathbf{R}^\top \mathbf{R} = \mathbf{R}\mathbf{R}^\top = \mathbf{I}, |\mathbf{R}| = 1 \Big\} \end{aligned} \tag{5}$$

In $SE(n)$ group, the movement of a rigid body B in Figure 4 can be explained by reference frame $\{A\}$ by creating another reference frame $\{B\}$ on B and describing the position and direction of B in relation to A using a homogeneous transformation matrix [19].

$${}^{A}\mathbf{A}_{B} = \left[\begin{array}{c:c} {}^{A}\mathbf{R}_{B} & {}^{A}\mathbf{t}^{O'} \\ \hdashline 0_{1\times n} & 1 \end{array}\right] \tag{6}$$

where ${}^{A}\mathbf{t}^{O'}$ is the translation vector of the origin O' of {B} in the reference frame $\{A\}$, and ${}^{A}\mathbf{R}_{B}$ is a rotation matrix that transforms the components of vectors in $\{B\}$ into components in $\{A\}$. Figure 4 presents an example of transformation from B to A which can be written as ${}^{A}\mathbf{t}^{P} ={}^{A} \mathbf{R}_{B}{}^{B}\mathbf{t}^{P'} +{}^{A} \mathbf{t}^{O'}$ in 3-dimensional Euclidean space. Moreover, the composition of two displacements, from $\{A\}$ to $\{B\}$, and from $\{B\}$ to $\{C\}$, is equal to the matrix multiplication of ${}^{A}\mathbf{A}_{B}$ and ${}^{B}\mathbf{A}_{C}$. Equation 7 illustrates the decomposition of the transformation $\{C\}$ to $\{A\}$ into two sub-tranformations $\{C\}$ to $\{B\}$ and $\{B\}$ to $\{A\}$.

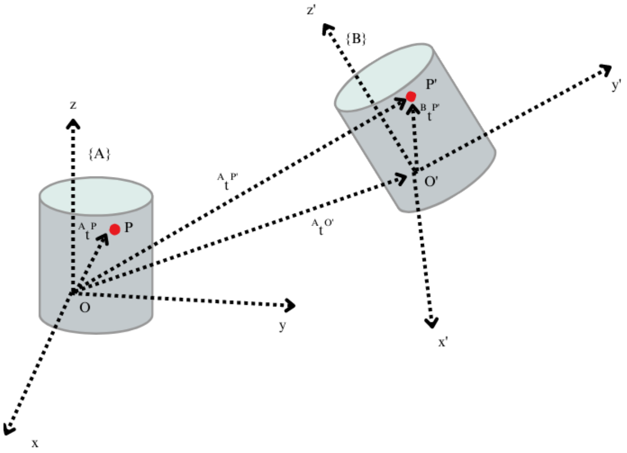

Figure 4: Example of rigid transformation in $SE(3)$.

$$\begin{aligned} {}^{A}\mathbf{A}_{C} &= \left[\begin{array}{c:c} {}^{A}\mathbf{R}_{C} & {}^{A}\mathbf{t}^{O''} \\ \hdashline 0_{1\times n} & 1 \end{array}\right] \\ &= \left[\begin{array}{c:c} {}^{A}\mathbf{R}_{B} & {}^{A}\mathbf{t}^{O'} \\ \hdashline 0_{1\times n} & 1 \end{array}\right] \times \left[\begin{array}{c:c} {}^{B}\mathbf{R}_{C} & {}^{B}\mathbf{t}^{O''} \\ \hdashline 0_{1\times n} & 1 \end{array}\right] \\ &= \left[\begin{array}{c:c} {}^{A}\mathbf{R}_{B} \times{}^{B} \mathbf{R}_{C} & {}^{A}\mathbf{R}_{B} \times{}^{B} \mathbf{t}^{O''} +{}^{A} \mathbf{t}^{O'} \\ \hdashline 0_{1\times n} & 1 \end{array}\right] \end{aligned} \tag{7}$$

It is evident from (7) that the transformation is reversible, meaning we can aggregate multiple transformations into one. Due to this property, assuming that the transformation ${}^{A}\mathbf{A}_{B}$ consists of a single rotation followed by a single translation, then $\exists{}^{A}\mathbf{A'}_{B} \in SE(n) \Rightarrow {}^{A}\mathbf{A'}_{B} = {}^{A}\mathbf{A}_{B}$.

III. Methodology

A. Overview of pipeline

Assume the task at hand is to identify the lexical mapping between two languages: a low-resource language and another language with a grammatical structure that exhibits similarity. In this context, the proposed method, referred to as ASK, functions as a comprehensive, end-to-end pipeline designed specifically to discover the mapping between the embedding spaces of the two languages. The ASK method is articulated into two primary phases, detailed as follows.

1) **Embedding Model Construction:** The initial phase involves constructing a unique embedding model for each language. For the low-resource language, two specific data augmentation techniques are employed to enhance the modeling process: Sentence Boundary Augmentation (SB) [20] and Multitask Learning Data Augmentation (MD) [21]. These techniques aim to improve the representational capacity of the embeddings, especially when dealing with limited data availability.
2) **Fine-tuning and Mapping Computation:** In the subsequent phase, the focus shifts to fine-tuning embedding models and computing the mapping between the embedding spaces of the two languages. A set of parallel words is randomly sampled from the collected bilingual dictionary and designated as anchor points. Utilizing the Kabsch algorithm, we fine-tune two embedding models for anchors to be aligned. Then, these anchors are employed to calculate the n-dimensional rigid transformation between the embedding spaces. This rigorous approach leverages the intrinsic geometric properties of the data, ensuring an accurate alignment of the linguistic structures.

B. Embedding model construction

In this study, we applied two below techniques to deal with data shortage of low-resource languages.

1) **Sentence Boundary Augmentation** is a noise-based approach at the sentence level. By truncating parts of sentences and then combining them, it can remove context from the first sentence, add context from the second sentence, and combine them into a single training example. The proportion of the sentences is governed by a hyperparameter. [20]
2) **Multitask Learning Data Augmentation** combines a set of simple data augmentation methods including Word Swap, Reverse, Semantic Embedding [22], Exploratory Data Analysis (EDA) [23] to produce synthetic sentences.

By adding noise to the text in this way, the embedding model can learn different embeddings for words based on the combination of sentences. These generated sentences along with the original ones are then used as the training data for learning monolingual embedding model [24], [25].

C. Fine-tuning and mapping computation with Kabsch algorithm

Firstly, we denote the real mapping between two languages as $f^*(\cdot)$ and the set of anchor words of these languages as $\mathcal{W}_A = \{w_i^A\}_{i=1}^N$ and $\mathcal{W}_B = \{w_i^B\}_{i=1}^N$ where $w_i^A = f^*(w_i^B)$. Considering the original embedding models for two languages are $\mathcal{M}_A$ and $\mathcal{M}_B$. We add linear transformations to the end of each model, thus, the embedding model should become $\mathcal{M}_A^\theta$, $\mathcal{M}_B^\gamma$ where θ and γ are learnable parameters. Then the vector sets of anchor words can be expressed as (8).

$$
\begin{aligned}
X^\gamma &= \{x_i = \mathcal{M}_B^\gamma(w_i^B) \in \mathbb{R}^n\}_{i=1}^N \\
Y^\gamma &= \{y_i = \mathcal{M}_A^\theta(w_i^A) \in \mathbb{R}^n\}_{i=1}^N
\end{aligned}
\tag{8}
$$

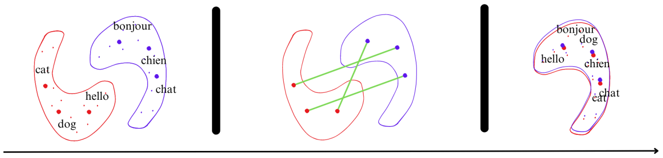

Figure 5: Example transformation with Kabsch algorithm

In this study, we treat the problem of finding mapping between two embedding spaces as Procrustes superimposition problem [**?**]. Therefore, we utilize the Kabsch algorithm to find the mapping or the transformation between two embedding point cloud, mathematically speaking. The objective of Kabsh algorithm is computing an approximation $f(\cdot)$ of the mapping $f^*(\cdot)$ to optimize the objective function in (9).

$$
f = \text{argmin}_f \mathbb{E}_{\substack{X \sim B \\ Y \sim A}} \left[||f(X) - Y||^2 \right] \tag{9}
$$

However, we can not directly optimize (9), so that we reparameterize it with θ and γ. The new objective function is then become (10). This objective function is also the loss function for fine-tuning embedding models.

$$
\mathcal{L} = \text{argmin}_{\theta,\gamma} \mathbb{E}_{\substack{X^\gamma \sim B \\ Y^\theta \sim A}} \left[||f(X^\gamma) - Y^\theta||^2 \right] \tag{10}
$$

Base on the theory of $SE(n)$ group, the $f(\cdot)$ represents an affine linear function: $\mathbb{R}^n \to \mathbb{R}^n$, which corresponds to a rigid motion in $\mathbb{R}^n$. Under the perspective of linear algebra, $f(x) = \mathbf{R}x + \mathbf{t}$ with $x \in \mathbb{R}^n$, where $\mathbf{R} \in \mathbb{R}^{n\times n}$, $|\mathbf{R}| = 1$, and $\mathbf{t} \in \mathbb{R}^n$. Nextly, we denote the centroid if point cloud X and Y in Equation 11.

$$
\begin{aligned}
\mu_X &= \frac{1}{N} \sum_{x_i \in X} x_i \\
\mu_Y &= \frac{1}{N} \sum_{y_i \in Y} y_i
\end{aligned}
\tag{11}
$$

The Kabsch algorithm is summarized in Algorithm 1. Figure 5 illustrates the transformation with Kabsch algorithm.

Algorithm 1 Kabsch algorithm

```
Input: Point cloud set $X, Y \in \mathbb{R}^{n\times N}$
Output: $\mathbf{R} \in \mathbb{R}^{n\times n}$, $\mathbf{t} \in \mathbb{R}^n$
  $C = XY^\top$
  Perform SVD: $C = U\Sigma V^\top$
  $\Sigma' = \{\sigma_i\}_{i=1}^n$, where $\sigma_{i<n} = 1$ and $\sigma_n = \text{sign}(|VU^\top|)$
  $\mathbf{R} = V\Sigma' U^\top$
  $\mathbf{t} = \mu_Y - \mathbf{R}\mu_X$
  return $\mathbf{R}, \mathbf{t}$
```

Table I: Number of sentences in Vietnamese and Bahnaric corpora

Dataset	Original	Augmented
Vietnamese	16105	78307
Bahnaric	16105	78307

Table II: Examples of French-to-English on 10000 anchors

Source Word	Top1	Top2	Top3	Top4
soins	**care**	deal	fear	attention
fin	**end**	close	goal	stop
chaque	**each**	apiece	vice	canso
position	**position**	place	emplacement	location
accès	**access**	accession	approach	admission
ouest	**west**	westward	eastern	easterly
période	**period**	stop	point	flow
emplois	**jobs**	job	subcontract	line
impôt	**tax**	taxes	taxation	assess
rôle	**role**	persona	character	function

After the embedding models are fine-tuning, we calculate the approximate mapping function using the same procedure. Consequently, the process of identifying the mapping of a source language word in the target language involves ranking the neighboring embedding points based on cosine similarity. Cosine similarity is a widely used metric in natural language processing that measures the similarity between two vectors in a high-dimensional space. By employing this approach, we can effectively determine the closest matching target language word or its nearest neighbors in the embedding space.

Next, we present the proof of better performance of the Kabsch algorithm in n-dimentional space in comparison to the original Procrustes problem and the SOTA approach.

a) Ensuring rigid transformation: Assuming that the objective of Procrustes problem is hold, denoted as (12).

$$\begin{aligned} g &= \text{argmin}_g \mathbb{E}_{\substack{X\sim A\\Y\sim B}}\Big[||g(X)-Y||^2\Big], g \in O(n) \\ &= \text{argmin}_g \mathbb{E}_{\substack{X\sim A\\Y\sim B}}\Big[\text{tr}\left((\mathbf{R}X-Y)^\top(\mathbf{R}X-Y)\right)\Big] \\ &= \text{argmin}_g \mathbb{E}_{\substack{X\sim A\\Y\sim B}}\Big[\text{tr}(X^\top X)+\text{tr}(Y^\top Y)-2\text{tr}(Y^\top \mathbf{R}X)\Big] \\ &= \text{argmax}_g \mathbb{E}_{\substack{X\sim A\\Y\sim B}}\Big[\text{tr}(Y^\top \mathbf{R}X)\Big] \\ &= \text{argmax}_g \mathbb{E}_{\substack{X\sim A\\Y\sim B}}\Big[\text{tr}(\mathbf{R}XY^\top)\Big] \end{aligned} \tag{12}$$

Let $C = XY^\top = U\Sigma V^\top$, since $V^T\mathbf{R}U$ is orthogonal, then

$$\text{tr}(\mathbf{R}C) = \text{tr}(\mathbf{R}U\Sigma V^T) = \text{tr}(V^T\mathbf{R}U\Sigma) \leq \text{tr}(\Sigma) = \sum_{j=1}^{n}\sigma_j. \tag{13}$$

The equality holds if $\mathbf{R} = VU^\top$ and $|VU^\top| > 0$. However, in case $|VU^\top| < 0$, the (13) becomes (14).

$$\text{tr}(\mathbf{R}C) = \text{tr}(\mathbf{R}U\Sigma V^T) = \text{tr}(V^T\mathbf{R}U\Sigma) \leq \sum_{j=1}^{n-1}(\sigma_j - \sigma_n). \tag{14}$$

Table III: Examples of Bahnaric-to-Vietnamese on 500 anchors

Source	Top1	Top2	Top3	Top4
máu	**pham**	thăm	chăn	tâng_kơjung
sữa	**đak_toh**	dư_dư	bek_bŏ	hla_piêt_yĕr
yên	**an**	krŭ	kơpung	areh
trôi	**đơng**	pơdrăn	prah	kơnăr
gì	**kiơ**	kŏ_jong	tơtuanh	bok_y
vỡ	**pơchah**	brôm	apăl_asơl	kơkŏch
bay	**apăl**	srang	bưp_bưp	long_wăk
tỏa	**tơprah**	chă_hming	hla_piêt_yĕr	bluh_lêch
thiếu	**bĭ_mah**	mơng_kơtang	ping_ngil	hming_ji
công	**kơwơng**	dư_dư	yĕr_tơmông	ngưk_ich

If we keep $|\mathbf{R}| = VU^\top$, we still achieve the equality but $|\mathbf{R}| = -1$ which causes the reflections in the original point cloud, which is not what we expect since we assume that the two sets of point cloud have the same shape. The Kabsch algorithm resolves this issue and get $g \in SO(n)$ by choosing $\mathbf{R} = V\Sigma' U^\top$, where $\Sigma' = \{\sigma_{i<n} = 1, \sigma_n = -1\}$.

b) Tackling translation in high-dimensional space: Assuming that we already solve the original Procrustes problem and get the mapping function $g(\cdot)$, we define our mapping function $f(\cdot)$ as (15).

$$f(X) = g(X) - g(\mu_X) + \mu_Y \tag{15}$$

Considering the difference between original solution and Kabsch algorithm as in (16), we observe that when $g(\mu_X) \neq \mu_Y$, the Kabsch algorithm, that takes translation into account will be convergence to the maxima while the original one can not.

$$\begin{aligned} \Delta &= ||g(X)-Y||^2 - ||f(X)-Y||^2 \\ &= \sum_{i=1}^{n}(\mathbf{R}x_i - y_i) - \sum_{i=1}^{n}(\mathbf{R}x_i - \mathbf{R}\mu_X + \mu_Y - y_i) \\ &= \sum_{i=1}^{n}(\mathbf{R}\mu_X - \mu_Y) \\ &= n||g(\mu_X) - \mu_Y||^2 \geq 0 \end{aligned} \tag{16}$$

IV. Experiments and discussions

In this section, we conduct a comprehensive comparison of our proposed approach with other baseline methods across various benchmarks. Our experimental analysis consists of two distinct phases. Firstly, we concentrate on well-resourced language pairs, particularly French-English, to showcase the effectiveness and efficiency of our method. Secondly, we extend our evaluation to the Vietnamese-Bahnaric language pair, strategically chosen to assess and verify our method performance in a setting with limited linguistic resources. This two-phase evaluation enables a robust examination of the generalizability and adaptability of our approach across different language scenarios, contributing to a deeper understanding of its capabilities and limitations.

A. Experimental setups

Toward experiments on rich-resource datasets, French-English, we uses a French-English corpus containing 53,241 words. We will train embeddings with three options:

Table IV: The comparison between Kabsch and the other supervised models on French-English

Method	**Top-1Acc**(%)↑	**Top-5Acc**(%)↑	**Top-10Acc**(%)↑	**MRR**↑	**Runtime**(ms)↓
1000 anchor words					
Artetxem	3.678 ± 0.289	7.094 ± 0.419	8.842 ± 0.461	0.05478 ± 0.0034	3819.0170 ± 99.1973
Dino	1.386 ± 0.198	3.353 ± 0.387	4.595 ± 0.474	0.02542 ± 0.00285	7.7471 ± 4.1768
Mikolov	1.388 ± 0.196	3.342 ± 0.385	4.584 ± 0.462	0.02537 ± 0.00283	2535.6096 ± 57.1759
Kabsch	3.984 ± 0.251	7.41 ± 0.375	9.066 ± 0.406	0.05779 ± 0.00301	$\mathbf{1.3448 \pm 0.2295}$
ASK	$\mathbf{19.14 \pm 0.217}$	$\mathbf{25.32 \pm 0.323}$	$\mathbf{27.13 \pm 0.354}$	$\mathbf{0.2056 \pm 0.001}$	1.4288 ± 0.2135
10000 anchor words					
Artetxem	7.812 ± 0.194	11.926 ± 0.295	13.679 ± 0.329	0.09909 ± 0.00238	3972.3056 ± 199.4734
Dino	1.886 ± 0.122	4.25 ± 0.195	5.674 ± 0.227	0.03233 ± 0.00157	25.0557 ± 0.5525
Mikolov	1.887 ± 0.109	4.256 ± 0.175	5.686 ± 0.198	0.03239 ± 0.00136	2524.3594 ± 23.2422
Kabsch	9.088 ± 0.054	13.547 ± 0.069	15.438 ± 0.082	0.1135 ± 0.0006	$\mathbf{2.7956 \pm 0.2618}$
ASK	$\mathbf{46.25 \pm 0.032}$	$\mathbf{53.19 \pm 0.035}$	$\mathbf{55.5 \pm 0.042}$	$\mathbf{0.4787 \pm 0.0014}$	$3.2143 \pm 0.0.1538$
50000 anchor words					
Artetxem	10.361 ± 0.508	15.369 ± 0.592	17.603 ± 0.431	0.1294 ± 0.00481	4538.6975 ± 53.7677
Dino	1.867 ± 0.195	4.141 ± 0.307	5.564 ± 0.302	0.03185 ± 0.0016	259.4690 ± 2.9341
Mikolov	1.922 ± 0.179	4.172 ± 0.306	5.5659 ± 0.288	0.03209 ± 0.00156	13438.7266 ± 93.4700
Kabsch	9.719 ± 0.414	14.07 ± 0.476	15.89 ± 0.51	0.11926 ± 0.0043	$\mathbf{9.6051 \pm 0.9784}$
ASK	$\mathbf{61.71 \pm 0.396}$	$\mathbf{66.34 \pm 0.413}$	$\mathbf{69.423 \pm 0.442}$	$\mathbf{0.6312 \pm 0.0036}$	$10.1524 \pm 0.1.1226$

1) 1,000 anchor words along with 52,241 test words.
2) 10,000 anchor words along with 43,241 test words.
3) 50,000 anchor words along with 3,241 test words.

For a fair model comparison, we use the rich-resource dataset without augmentation. Synonyms of English words are found using WordNet from Princeton University [26] and implemented by NLTK[1] for evaluation.

Furthermore, we will assess the impact of data augmentation on our low-resource datasets through two different tests:

1) Evaluation using the original datasets.
2) Evaluation using augmented data from the original dataset, which includes sentences with sentence boundaries, EDA, and semantic embedding augmentation combined with the original datasets.

The dataset information, comprising both the original data and its augmented counterpart, is provided in Table I. The original dataset is represented in the 'Original' column, while the augmented dataset is found in the 'DA' column.

The embeddings will be trained with three options:

1) 100 anchor words along with the rest being test words.
2) 500 anchor words along with the rest being test words.
3) 1000 anchor words along with the rest being test words.

During training, ASK utilizes Singular Value Decomposition (SVD) for learning the mapping, and no hyperparameters are required. However, the word embeddings also play a critical role. After conducting multiple experiments, we selected the Skip-gram model to learn the word embeddings with the following settings: the hidden dimension is 100, the window size is 5, and words whose frequency less than 2 are ignored.

We have employed two commonly used metrics which are listed in the followings to evaluate the ranking performance of our model.

1) Mean Reciprocal Rank (MRR): This metric incorporates synonyms in addition to exact word matching. By considering synonyms, we obtain a more comprehensive evaluation of the mapping quality. To evaluate the model, we compute the mean MRR across all testing words.
2) Top-K accuracy (Top-KAcc): This metric evaluates the model performance by examining the Top-K ranked results and assessing the position of the correct word.
3) Runtime: This metric quantifies the elapsed time taken by the model to identify the mapping function responsible for translating source language words to their corresponding target language words.

To improve performance on low-resource datasets, we employ a fine-tuning strategy. Our model consists of three linear layers that project the original embeddings into a shared space, ensuring that both the source and target mapped embeddings have the same shape. We use hidden state dimensions are set to 1024 and 2048 and activate these layers using Relu and Tanh functions, as they yielded the best results during experimentation. The training process maintains a constant learning rate of 10^{-3} across dataset sizes (100, 500, 1000) but extends the number of epochs (20000, 40000, 80000) for enhanced optimization. Our chosen optimization method is Stochastic Gradient Descent (SGD).

B. Baselines

The study of Mikolov [13] utilizes skip-gram word embedding to learn high-quality word embeddings, opting for a rotation matrix that minimizes the loss function $sum_{i=1}^{n}||Wx_i - z_i||^2$. By employing gradient descent, they find optimal values for the matrix W, enabling seamless mapping between the word spaces of source and target languages without constraints. The authors then identify the target language word with the highest cosine similarity to z, establishing meaningful

[1] https://www.nltk.org/

Table V: The comparison between our method, its ablated versions (with fine-tuning (FT) and data augmentation (DA)) and the other supervised models on Vietnamese-Bahnaric

Method	**Top-1Acc**($\%$)↑	**Top-5Acc**($\%$)↑	**Top-10Acc**($\%$)↑	**MRR**↑	**Time**(ms)↓
100 anchor words					
Artetxem	0.8 ± 1.5	1.2 ± 1.7	1.5 ± 1.7	0.012 ± 0.016	0.641 ± 0.122
Dino	0.1 ± 0.1	0.6 ± 0.2	1.3 ± 0.3	0.008 ± 0.001	0.015 ± 0.003
Mikolov	4.9 ± 0.1	5.3 ± 0.1	5.5 ± 0.2	0.054 ± 0.001	2.450 ± 0.122
Kabsch	1.3 ± 0.1	2.2 ± 0.2	3.1 ± 0.3	0.022 ± 0.001	$\mathbf{0.001 \pm 0.0003}$
Kabsch + FT	5.0 ± 0.1	5.3 ± 0.1	5.6 ± 0.3	0.054 ± 0.001	0.0033 ± 0.0001
Kabsch + DA	2.9 ± 0.2	3.8 ± 0.3	4.4 ± 0.5	0.037 ± 0.003	0.001 ± 0.0003
ASK	$\mathbf{6.0 \pm 0.04}$	$\mathbf{6.1 \pm 0.05}$	$\mathbf{6.4 \pm 0.1}$	$\mathbf{0.063 \pm 0.0004}$	0.0035 ± 0.016
500 anchor words					
Artetxem	9.9 ± 0.5	13.3 ± 0.5	15.2 ± 0.5	0.119 ± 0.004	0.679 ± 0.127
Dino	0.2 ± 0.1	1.0 ± 0.3	2.0 ± 0.4	0.012 ± 0.001	0.015 ± 0.004
Mikolov	6.0 ± 0.3	7.5 ± 0.3	8.4 ± 0.3	0.071 ± 0.002	2.567 ± 0.054
Kabsch	3.2 ± 0.5	4.8 ± 0.6	6.0 ± 0.9	0.044 ± 0.005	$\mathbf{0.001 \pm 0.0001}$
Kabsch + FT	30.4 ± 0.1	30.6 ± 0.2	30.8 ± 0.2	0.306 ± 0.001	0.0037 ± 0.0001
Kabsch + DA	8.3 ± 0.4	10.4 ± 0.6	11.8 ± 0.8	0.097 ± 0.004	0.001 ± 0.0001
ASK	$\mathbf{33.3 \pm 0.1}$	$\mathbf{33.5 \pm 0.1}$	$\mathbf{33.7 \pm 0.1}$	$\mathbf{0.336 \pm 0.001}$	0.0038 ± 0.0001
1000 anchor words					
Artetxem	11.9 ± 0.8	16.7 ± 0.7	19.0 ± 0.6	0.145 ± 0.007	0.685 ± 0.164
Dino	0.2 ± 0.1	0.99 ± 0.3	1.9 ± 0.4	0.012 ± 0.002	0.017 ± 0.003
Mikolov	8.2 ± 0.8	9.5 ± 0.6	10.5 ± 0.7	0.093 ± 0.007	2.540 ± 0.028
Kabsch	4.6 ± 0.5	6.7 ± 0.6	8.1 ± 0.5	0.061 ± 0.005	0.001 ± 0.0001
Kabsch + FT	59.1 ± 0.8	60.0 ± 0.3	60.4 ± 0.3	0.597 ± 0.005	0.004 ± 0.0001
Kabsch + DA	11.1 ± 0.6	14.1 ± 0.8	16.2 ± 1.0	0.131 ± 0.007	$\mathbf{0.001 \pm 0.0002}$
ASK	$\mathbf{64.9 \pm 0.2}$	$\mathbf{65.0 \pm 0.1}$	$\mathbf{65.0 \pm 0.1}$	$\mathbf{0.650 \pm 0.001}$	0.004 ± 0.00035

associations between words in different languages for cross-lingual tasks like translation and word alignment.

The Mikolov model [13] lacks constraints, which may lead to overfitting and underutilization of word embedding features. To address this, the Dinu model [27] introduces regularization to prevent specific words from being consistently mapped to particular targets. Additionally, they modify the method for selecting the correct word after mapping the source language word using the matrix W. This change is necessary because cosine similarity, commonly used for this task, encounters the Hubness problem—an inherent challenge in high-dimensional spaces [28] and a recognized issue for word-based vectors [28]. As a result, their focus lies on proposing a straightforward and efficient solution to handle this problem by adjusting the similarity matrix post-mapping process.

And the last model which we use for comparing our result is Artetxe model [29]. Their method is remarkable for its effectiveness even with just 25 word pairs, a departure from previous methods that often require thousands of words for satisfactory performance. They emphasize the adaptability of their approach with low-dimensional pre-trained word embeddings. For inducing bilingual lexicons, a common evaluation task, they use a small train set (seed dictionary) to learn an initial mapping, leading to a larger and potentially enhanced dictionary. In the second step, they train the model to refine the source-to-target language mapping, aiming for improvements over the input dictionary. This iterative process allows for continuous refinement until a convergence criterion is met.

C. Evaluations using rich-resource datasets

This experiment assesses the effectiveness of Kabsch algorithm, in finding language mappings between French and English datasets (rich-resource datasets) with similar point cloud shapes. The analysis (Table IV) demonstrates that Kabsch outperforms most other methods when utilizing 1000 and 10,000 anchor points. However, Our ASK model outperforms other methods due to its fine-tuned embedding, which aligns the shapes of the source and target language embeddings.

Nonetheless, Kabsch consistently achieves favorable results across all cases, maintaining a relatively lower runtime compared to other methods. Kabsch exhibits the lowest runtime among the tested models, making it a promising approach for efficient and accurate language mapping tasks. To showcase the mapping process, we have randomly chosen 10 words, which are presented in Table II. Each "Top i" column representing the i^{th} target word with the highest similarity score.

D. Evaluations using low-resource datasets

In this scenario, we executed full pipeline of ASK including data augmentation, fine-tuning embedding models and computing mapping with Kabsch. We compared our method with its ablated versions and other supervised learning models in terms of Top-K Accuracy and mapping computation runtime for Vietnamese-Bahnaric in the Table V. Furthermore, we have randomly selected 10 Vietnamese words to illustrate the mapping process. These words are displayed in Table III with the same column meanings as in Table II.

1) Evaluations using original datasets: In general, when Kabsch is used with low-resource data, its outcomes tend to be slightly less impressive compared to alternative models. This can be traced back to the data's limited scale. Since the dataset is small, it might fail to meet the criteria for the embedding shapes to match exactly, resulting in a decline in accuracy. However, by implementing Finetuning on these embeddings, accuracy improves more effectively than relying solely on Kabsch. It's important to highlight that Kabsch's runtime has been notably reduced compared to other techniques. Kabsch emerges as the fastest performer in terms of execution speed.

2) Evaluations using augmented datasets: Through the application of various augmentation techniques such as sentence boundary augmentation, EDA, and word2vec on the original dataset, we have significantly expanded its size, nearly multiplying it by a factor of 8. As a result, we have observed a considerable improvement in performance when compared to evaluating the model on the original data alone. This enhancement arises from the model's improved ability to learn the underlying distribution of the data. Notably, our proposed method achieves higher Top-1 Accuracy and MRR scores in comparison to alternative approaches. This observation underscores the advantage of employing a larger dataset and highlights the fulfillment of the underlying assumption. These factors contribute to the superior performance of our approach compared to other methods, all while maintaining a lower runtime.

V. Conclusion

This paper introduces a novel approach for word alignment based on distribution representations. Leveraging two monolingual language corpora and an initial dictionary, our method effectively learns a meaningful transformation for individual words. The experimental results reveal the efficacy of our approach on rich-resource datasets, exhibiting superior training time compared to alternative methods. Additionally, promising performance is observed on low-resource datasets, highlighting the potential for broader applicability.

In the future, we intend to conduct further investigations in this direction, aiming to refine and optimize our method to ensure a more coherent shape for word embeddings from two monolingual language corpora. This enhancement will facilitate more efficient alignment between the corpora, ultimately leading to improved alignment accuracy and precision. Our ongoing research aims to enhance the practicality and versatility of our approach, enabling cross-lingual language processing and effective multilingual resource alignment.

Acknowledgment

This research is funded by Ministry of Science and Technology (MOST) within the framework of the Program "Supporting research, development and technology application of Industry 4.0"KC-4.0/19-25 – Project “Development of a Vietnamese- Bahnaric machine translation and Bahnaric text-to-speech system (all dialects)” - KC-4.0-29/19-25.

List of acronyms

AI	Artificial Intelligence
GANs	Generative Adversarial Networks
EDA	Exploratory Data Analysis
ASK	**A**ugmenting and **S**ampling with **K**absch
MRR	Mean Reciprocal Rank
SVD	Singular Value Decomposition
SOTA	state-of-the-art

Conflict of interest

The authors hereby declare that there is no conflict of interest in the publication of this article.

Author contribution statement

- La Cam Huy: Gathering data in English and French, performing preprocessing on data in English, French, Vietnamese, and Bahnaric languages, searching for relevant problem-solving models, constructing models, comparing results, and writing research papers.
- Le Quang Minh: Collecting information in English and French, organizing information in English, French, Vietnamese, and Bahnaric languages, finding problem-solving methods that are related to the topic, and writing research papers.
- Tran Ngoc Oanh: Performing preprocessing on data in English, French, Vietnamese and Bahnaric languages. Augmenting the dataset and writing research papers
- Le Duc Dong: Augmenting the dataset, supporting model construction, writing the research paper
- Duc Q. Nguyen: Come up with ideas for writing articles, collect data in English, French, Vietnamese and Bahnaric. Testing models, tutorials and editing paper.
- Nguyen Tan Sang: Participate in the extending data for Vietnamese and Bahnaric.
- Tran Quan: Participate in coming up writing ideas
- Tho Quan: Come up with ideas for writing articles, collecting data in Vietnamese, Bahnaric. Providing paper tutorials and editing.

References

[1] W. Zhu, Z. Zhou, S. Huang, Z. Lin, X. Zhou, Y. Tu, and J. Chen, “Improving bilingual lexicon induction on distant language pairs,” in *Machine Translation*, S. Huang and K. Knight, Eds. Singapore: Springer Singapore, 2019, pp. 1–10.

[2] C. Janiesch, P. Zschech, and K. Heinrich, “Machine learning and deep learning,” *Electronic Markets*, vol. 31, no. 3, p. 685–695, Apr 2021.

[3] S. K. Mondal, H. Zhang, H. M. D. Kabir, K. Ni, and H.-N. Dai, “Machine translation and its evaluation: a study,” *Artificial Intelligence Review*, vol. 56, no. 9, p. 10137–10226, Feb 2023.

[4] G. Lample, A. Conneau, M. Ranzato, L. Denoyer, and H. Jégou, “Word translation without parallel data,” in *International Conference on Learning Representations*, 2018.

[5] C. Tang, X. Yang, B. Wu, Z. Han, and Y. Chang, “Parts2words: Learning joint embedding of point clouds and texts by bidirectional matching between parts and words,” in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2023, pp. 6884–6893.

[6] I. M. Yaglom, *Geometric transformations*. Washington: Mathematical Association of America,, 1962.

[7] H. W. Guggenheimer, *Plane geometry and its groups*. Cambridge University Press, 1968, vol. 11, no. 3, p. 508–509.

[8] V. G. Satorras, E. Hoogeboom, and M. Welling, “E(n) equivariant graph neural networks,” in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 9323–9332.

[9] R. Rubino, B. Marie, R. Dabre, A. Fujita, M. Utiyama, and E. Sumita, “Extremely low-resource neural machine translation for asian languages,” *Machine Translation*, vol. 34, no. 4, p. 347–382, Dec 2020.

[10] Z. Li, P. Xia, R. Tao, H. Niu, and B. Li, “A new perspective on stabilizing gans training: Direct adversarial training,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 1, pp. 178–189, 2023.

[11] S. Wang, Y. Yang, Z. Wu, Y. Qian, and K. Yu, “Data augmentation using deep generative models for embedding based speaker recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2598–2609, 2020.

[12] K. K. Gupta, S. Sen, R. Haque, A. Ekbal, P. Bhattacharyya, and A. Way, “Augmenting training data with syntactic phrasal-segments in low-resource neural machine translation,” *Machine Translation*, vol. 35, no. 4, p. 661–685, Dec 2021.

[13] T. Mikolov, Q. V. Le, and I. Sutskever, “Exploiting similarities among languages for machine translation,” 2013.

[14] C. Zhou, X. Ma, D. Wang, and G. Neubig, “Density matching for bilingual word embedding,” in *North American Chapter of the Association for Computational Linguistics*, 2019.

[15] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, “Unsupervised cross-lingual representation learning at scale,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 8440–8451.

[16] C. Xing, D. Wang, C. Liu, and Y. Lin, “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*. Denver, Colorado: Association for Computational Linguistics, May–Jun. 2015, pp. 1006–1011.

[17] W. Ammar, G. Mulcaire, Y. Tsvetkov, G. Lample, C. Dyer, and N. A. Smith, “Massively multilingual word embeddings,” 2016.

[18] E. Artin, *Geometric Algebra*. John Wiley & Sons, 2011.

[19] H. S. M. Coxeter and S. L. Greitzer, *Geometry Revisited*. Mathematical Association of America, 2016.

[20] D. Li, T. I, N. Arivazhagan, C. Cherry, and D. Padfield, “Sentence boundary augmentation for neural machine translation robustness,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 7553–7557.

[21] E. Meyerson and R. Miikkulainen, “Pseudo-task augmentation: From deep multitask learning to intratask sharing—and back,” in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 739–748.

[22] W. Y. Wang and D. Yang, “That’s so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 2557–2563.

[23] J. W. Wei and K. Zou, “EDA: easy data augmentation techniques for boosting performance on text classification tasks,” *CoRR*, vol. abs/1901.11196, 2019.

[24] E. Voita, R. Sennrich, and I. Titov, “Analyzing the source and target contributions to predictions in neural machine translation,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 1126–1140.

[25] D. Dong, H. Wu, W. He, D. Yu, and H. Wang, “Multi-task learning for multiple language translation,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 1723–1732.

[26] Princeton University, “About wordnet,” 2010.

[27] G. Dinu and M. Baroni, “Improving zero-shot learning by mitigating the hubness problem,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[28] M. Radovanovic, A. Nanopoulos, and M. Ivanovic, “Hubs in space: Popular nearest neighbors in high-dimensional data,” *Journal of Machine Learning Research*, vol. 11, no. sept, pp. 2487–2531, 2010.

[29] M. Artetxe, G. Labaka, and E. Agirre, “Learning bilingual word embeddings with (almost) no bilingual data,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 451–462.

AUTHOR BIOGRAPHY

a) La Cam Huy: was born in the District 5 of Ho Chi Minh City in the year 2003. In 2021, he embarked on his educational journey at Ho Chi Minh City University of Technology (HCMUT), where he enthusiastically pursued his passion for computer science.

In 2022, he have become an research assistants with VNPT Labs, situated within the precincts of Ho Chi Minh City University of Technology (HCMUT). Currently, he is in the midst of his third year at HCMUT.

b) Le Quang Minh: was born in Dong Thap Province in 2002. In 2020, he started studying at Ho Chi Minh City University of Technology (HCMUT), where he happily followed his interest in computer science.

In 2022, he started working as a research assistant at VNPT Lab, which is located at HCMUT. Right now, he is in his third year at HCMUT.

c) Tran Ngoc Oanh: was born in the District 1 of Ho Chi Minh City in 2002. In 2020, she started her journey at Ho Chi Minh City University of Technology (HCMUT), where she got opportunities to pursue her passion for computer science.

In 2022, she started working as an research assistants at VNPT Labs, situated within the precincts of Ho Chi Minh City University of Technology (HCMUT). She is currently a fourth year student at HCMUT

d) Le Duc Dong: is a motivated student of computer science who has a passion for machine learning and natural language processing (NLP). He is currently in his fourth year of study at Ho Chi Minh City University of Technology, where he has taken courses on artificial intelligence, data mining, natural language processing.

e) Duc Q. Nguyen: recently graduated with a Bachelor of Engineering in Computer Science. Now he is working as Research Assistant, Teaching Assistant and Researcher under the supervision of Assoc. Prof. Tho Quan. His research interests include Artificial Intelligence, Computational Biology and Graph Representation Learning. His biggest ambition is to make human life better and better using his talent and experiences.

f) Nguyen Tan Sang: Sang received his bachelor’s degree in computer science and is currently pursuing a master’s degree in computer science at Ho Chi Minh City University of Technology, VNU-HCM, Vietnam. During his studies, Sang focused on research and problem-solving in areas such as web design, IoT, machine learning, and machine translation. He has also dedicated for 3 years to contributing to a social network platform called MetaFox.

g) Tran Quan: Quan was a former graduate of Ho Chi Minh City University of Technology (HCMUT), obtaining a master's degree with a thesis about Natural Language Processing in 2021. Currently, he holds the position of Senior Executive at Retailers AI, offering business intelligence solutions for the retail industry.

h) Tho Quan: Dr. Quan Thanh Tho is an Associate Professor, currently working at Computer Science and Engineering, University of Science and Technology - Vietnam National University, Ho Chi Minh City (HCMUT). He graduated with a bachelor's degree from HCMUT in 1998 and received a PhD in science from Nanyang Technological University, Singapore in 2002.

Currently, he is the Vice Dean of the Faculty and the head of the Computer Science program (Undergraduate level). Besides teaching, Dr. Tho is also a Scientific Advisor of the Research and Development (R&D) segment of YouNet Group since 2011. As a Scientific Advisor, he has Lead research and development activities for new products that meet the company's market needs using natural language processing and machine learning techniques.

Hướng tiếp cận thu giảm số chiều cho phép ánh xạ từ vựng tiếng Việt sang tiếng Ba Na từ các tập ngữ liệu không song song

La Cẩm Huy[*†], Lê Quang Minh[*†], Trần Ngọc Oanh[*†], Lê Đức Đồng[*†], Nguyễn Quang Đức[*†], Nguyễn Tấn Sang[*†], Trần Quân[*†], Quản Thành Thơ[*†‡]

[*]Trường Đại học Bách khoa, Đại học Quốc gia Thành phố Hồ Chí Minh, 268 Lý Thường Kiệt, Phường 14, Quận 10, Thành phố Hồ Chí Minh, Việt Nam

[†]Đại học Quốc gia Thành phố Hồ Chí Minh, Phường Linh Trung, Thành phố Thủ Đức, Thành phố Hồ Chí Minh, Việt Nam

[‡]Tác giả liên lạc: qttho@hcmut.edu.vn

Tóm tắt — Từ điển song ngữ là công cụ quan trọng cho việc dịch máy tự động. Bằng cách tận dụng các kỹ thuật học máy tiên tiến, chúng ta có thể xây dựng từ điển song ngữ bằng cách tự động học các sự ánh xạ từ vựng từ tập văn bản song ngữ. Tuy nhiên, việc thu thập tập văn bản song ngữ phong phú cho các ngôn ngữ ít tài nguyên, chẳng hạn như ngôn ngữ Ba Na, đặt ra một thách thức đáng kể. Những nghiên cứu gần đây cho thấy rằng các tập văn bản đơn ngữ, kết hợp với *từ neo* (anchor words), có thể hỗ trợ trong quá trình học các ánh xạ này. Phương pháp thường được áp dụng bao gồm sử dụng Mạng GAN (Generative Adversarial Networks) kết hợp giải quyết vấn đề *trực giao Procrustes* để tạo ra sự ánh xạ này. Phương pháp nàys thường không ổn ịnh và đòi hỏi tài nguyên tính toán đáng kể, đưa đến những khó khăn tiềm ẩn khi xử lý những ngôn ngữ ít tài nguyên như tiếng Ba Na được thu thập ở vùng sâu vùng xa. Để giảm thiểu điều này, chúng tôi đề xuất một chiến lược điều chỉnh *số chiều thấp* (low-rank), trong đó các hạn chế của GAN có thể được tránh bằng cách tính toán trực tiếp sự biến đổi giữa ngôn ngữ nguồn và ngôn ngữ đích. Chúng tôi đã đánh giá phương pháp của mình bằng cách sử dụng một bộ dữ liệu giàu tài nguyên giữa tiếng Pháp - tiếng Anh và một bộ dữ liệu ít tài nguyên giữa tiếng Việt – tiếng Ba Na. Đáng chú ý, sự ánh xạ từ vựng giữa tiếng Việt- tiếng Ba Na được tạo ra bằng phương pháp của chúng tôi có giá trị không chỉ trong lĩnh vực khoa học máy tính, mà còn đóng góp đáng kể vào việc bảo tồn di sản văn hóa của ngôn ngữ Ba Na trong cộng đồng dân tộc thiểu số của Việt Nam.

Từ Khoá — Thu giảm số chiều, ánh xạ từ vựng, ngôn ngữ ít tài nguyên, giải thuật Kabsch