Utilizing Language-Image Pretraining for Efficient and Robust Bilingual Word Alignment

Tuan Dinh*, Jy-yong Sohn, Shashank Rajput, Timothy Ossowski, Yifei Ming, Junjie Hu, Dimitris Papailiopoulos, Kangwook Lee
University of Wisconsin, Madison, WI, USA

Abstract

Word translation without parallel corpora has become feasible, rivaling the performance of supervised methods. Recent findings have shown the improvement in accuracy and robustness of unsupervised word translation (UWT) by utilizing visual observations, which are universal representations across languages. Our work investigates the potential of using not only visual observations but also pretrained language-image models for enabling a more efficient and robust UWT. We develop a novel UWT method dubbed Word Alignment using Language-Image Pretraining (WALIP), leveraging visual observations via the shared image-text embedding space of CLIPs (Radford et al., 2021). WALIP has a two-step procedure. First, we retrieve word pairs with high confidences of similarity, computed using our proposed *image-based fingerprints*, which define the initial pivot for the alignment. Second, we apply our *robust Procrustes algorithm* to estimate the linear mapping between two embedding spaces, which iteratively corrects and refines the estimated alignment. Our extensive experiments show that WALIP improves upon the state-of-the-art performance of bilingual word alignment for a few language pairs across different word embeddings and displays great robustness to the dissimilarity of language pairs or training corpora for two word embeddings.

1 Introduction

Translating words across different languages is one of the long-standing research tasks and a standard building block for general machine translation. Word translation is helpful for various downstream applications, such as sentence translation (Conneau et al., 2017; Hu et al., 2019) or cross-lingual transfer learning in language models (de Vries and Nissim, 2020). Unsupervised word translation (UWT) has recently drawn a great deal of attention (Artetxe et al., 2017; Conneau et al., 2017; Hartmann et al., 2019), reducing the need for bilingual supervision.

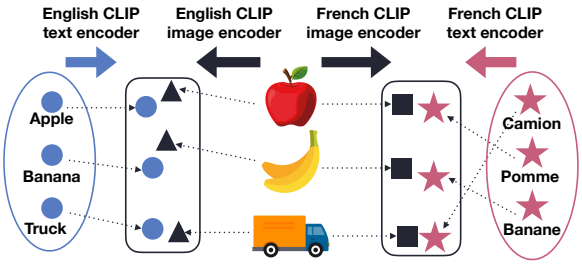


Figure 1: Conceptual visualization of WALIP for unsupervised word translation between English and French. We can connect English and French words in an unsupervised fashion through the shared images. CLIP models (Radford et al., 2021) can be used as human simulators to associate words with images.

Without any prior knowledge of the languages' connection, aligning their words is non-trivial. Most works on UWT exploit the structural similarity between continuous word embedding spaces across languages (Mikolov et al., 2013a; Ormazabal et al., 2019) to learn a linear mapping. Early works (Smith et al., 2017; Artetxe et al., 2017; Conneau et al., 2017; Hoshen and Wolf, 2018; Grave et al., 2019) focus on using only the text data to establish the bilingual alignment and solve the Procrustes problem (Schönemann, 1966). These methods rely on the similarity between pairs of languages and training corpora, thus not working well when the languages or corpora are dissimilar (Søgaard et al., 2018; Sigurdsson et al., 2020). They may also need a large amount of data to achieve good alignments (Sigurdsson et al., 2020).

Words can also be connected via the visual world. Visual similarity provides additional prior knowledge for easing language translation (Mihalcea and Leong, 2008). Recent works (Sigurdsson et al., 2020; Surís et al., 2020) demonstrate the promise of using visual information to improve UWT. However, they mostly require intense joint training for the embedding shared between images or videos with texts of multiple languages. More-

*Email: Tuan Dinh (tuan.dinh@wisc.edu)

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 154 - 168
December 7-11, 2022

over, these embeddings are used for translating all words, whereas not every word can be described by images or videos. Thus, it is unclear how they are helpful for non-visual words and whether the methods properly utilize topological similarity between word vector spaces (Mikolov et al., 2013a).

Our contributions. We propose WALIP (**W**ord **A**lignment with **L**anguage-**I**mage **P**retraining) as a new unsupervised word alignment method that leverages the joint image-text embeddings provided by CLIP (Radford et al., 2021). Fig. 1 shows an example inspiring WALIP. Consider a conversation between a French and an English speaker. As the English speaker shows an `apple` image, the French speaker can easily understand and provide its translation as `pomme`. They can similarly pair more words describing simple objects, helping translate more complex words. This observation inspires us to leverage visual information as the pivot for matching words across languages. To do so, we use CLIP (Radford et al., 2021) to correlate texts and images and construct an image-based word representation, called a *fingerprint*, where each coordinate measures the similarity between the word and an image a diverse image set. Note that fingerprints share similar merits with the pictorial representation of sentence (Mihalcea and Leong, 2008) that represents simple sentences by sequences of pictures. We use fingerprints to identify initial word pairs. As not every word can be described by images, we rely on the topological similarity of word vector spaces (Mikolov et al., 2013b) for the full alignment in the second step, *i.e.*, solving a linear mapping between two spaces using our robust Procrustes algorithm with identified word pairs.

Via extensive experiments, we show that WALIP is highly effective in bilingual alignment. We achieve comparable or better performance than the state-of-the-art (SOTA) baselines and close the gap to supervised methods. For instance, on the Dictionary benchmark (Sigurdsson et al., 2020) with HowToWorld-based word embedding (Miech et al., 2019), we achieve the SOTA performance on all evaluated pairs (English→{French, Korean, Japanese}), achieving significant accuracy improvements (6.7%, 2.5%, and 4.5%, *cf.* Table 2) over the previous SOTA (Sigurdsson et al., 2020). Our method also displays great robustness to the dissimilarity of language pairs and static word embeddings. We empirically show the effectiveness of our method through various ablation studies.

2 Related Works

Unsupervised word translation (UWT). Most UWT methods exploit the structure similarity between word vector spaces across languages (Mikolov et al., 2013a) to learn linear mappings. Early works (Smith et al., 2017; Artetxe et al., 2017) establish the parallel vocabulary and estimate the mapping by solving the Procrustes problem (Schönemann, 1966; Gower and Dijksterhuis, 2004). Others study assignment problems and directly solve Wasserstein-Procrustes for the one-to-one word assignment matrix (Zhang et al., 2017b; Grave et al., 2019) or hyper-alignment for multiple languages (Alaux et al., 2018; Taitelbaum et al., 2019). Recent works (Zhang et al., 2017a; Conneau et al., 2017; Hoshen and Wolf, 2018) propose to learn the mapping via aligning the embedding distributions with the notable MUSE framework (Conneau et al., 2017) using the adversarial training to achieve high translation performance for multiple pairs. We use MUSE as our baseline. While MUSE involves intense training for aligning two embedding spaces, WALIP does not require this training by utilizing pretrained CLIP models.

Visual information has been used to improve machine translation (Hewitt et al., 2018; Zhou et al., 2018; Kiros et al., 2018; Yang et al., 2020; Li et al., 2022b). Focusing on word translation, MUVE (Sigurdsson et al., 2020) trains a linear mapping between two embeddings via learning a joint video-text embedding space for pairs with captioned instructional videos. Globetrotter (Surís et al., 2020) learns the multilingual text embeddings aligned with image embeddings via contrastive learning. The learned text embeddings are used for multilingual sentence translation and refined for word translation. These methods require intense training with a large amount of vision-text data for learning the encoders, while WALIP only utilizes pretrained embeddings of off-the-shelf CLIP models. MUVE and Globetrotter are our main baselines.

Language-Vision (LV) models. We can categorize LV models into two types: single-stream and dual-stream models. The former feeds the concatenation of text and visual features into a single transformer-based encoder, such as VisualBERT (Li et al., 2019) and ViLT (Kim et al., 2021). The latter uses separate encoders for text and image and aligns semantically similar features in different modalities with contrastive objectives,

such as CLIP (Radford et al., 2021), ALIGN (Jia et al., 2021), and FILIP (Yao et al., 2021). We use CLIP as our language-image pretraining model due to its inference efficiency, high performance, and the availability of pretrained models in multiple languages. CLIP inspires numerous works (Zhang et al., 2021; Li et al., 2022a; Zhou et al., 2022) for better data efficiency and task adaptation of LV models. In this line of work, Zhai et al. (2022) recently show the feasibility of training multilingual image-text models without parallel corpora by connecting languages via image embeddings.

3 Problem Setup and Preliminaries

We formally describe the target problem of unsupervised word alignment and provide two preliminaries to our method: Procrustes and CSLS.

Unsupervised word alignment. We focus on the word alignment (translation) problem: finding the mapping from A_{dict} to B_{dict}, where $A_{\text{dict}} = \{a_1, \cdots, a_{n_a}\}$ and $B_{\text{dict}} = \{b_1, \cdots, b_{n_b}\}$ are dictionaries of source language A and target language B, with n_a and n_b being the number of words in each dictionary, respectively. This mapping can be represented by an equivalent index mapping $\pi : [n_a] \to [n_b]$, *i.e.*, we consider word a_i is mapped (aligned) to word $b_{\pi(i)}$, for $i \in [n_a]$. Here, $[n] = \{1, 2, \cdots, n\}$ is defined as the set of positive integers up to a positive number n. Note that we focus on *unsupervised* word alignment in which no ground-truth word pairs $(a_i, b_{\pi(i)})$ are given to the algorithm. To solve this problem, we assume the access to three ingredients: (1) a large-scale image dataset with d images denoted by $G = \{g_1, \cdots, g_d\}$, (2) a pre-trained monolingual CLIP model for each language, and (3) static word embeddings (Bojanowski et al., 2016; Pennington et al., 2014) for all words in dictionaries.

Procrustes problem. Let $X, Y \in \mathbb{R}^{n \times d}$ be matrices of the $d-$dimensional embeddings for n words in the source and target languages. The Procrustes problem aims to find $W \in \mathbb{R}^{d \times d}$ such that $\|XW - Y\|_F$ is minimized. Regularizing W with the orthogonality is found to improve the translation (Xing et al., 2015), where the optimal W is

$$W^* = \underset{W \in \mathcal{O}_d}{\texttt{argmin}} \|XW - Y\|_F = \texttt{SVD}(Y^T X)$$

where $\mathcal{O}_d$ is the set of $d \times d$ orthogonal matrices and SVD is the singular value decomposition.

CSLS. Conneau et al. (2017) proposed Cross-domain Similarity Local Scaling (CSLS) to robustly measure the similarity between words' embeddings. Given two sets $X = \{x_i\}_{i \in [n_X]}$, $Y = \{y_i\}_{i \in [n_Y]}$ and the number of neighbors K, the CSLS of x_i and y_j is defined as $\text{CSLS}(x_i, y_j) = 2\cos(x_i, y_j) - r_Y(x_i) - r_X(y_j)$ where $\cos(\cdot, \cdot)$ is the cosine similarity, $r_Y(x_i) = \frac{1}{K} \sum_{y_j \in \mathcal{N}_Y(x_i)} \cos(x_i, y_j)$ is the average similarity of x_i, and $\mathcal{N}_Y(x_i)$ is the set of K nearest neighbors of x_i among elements of Y. CSLS performs cross-domain normalization to address the hub phenomenon (Radovanovic et al., 2010) of the K-nearest-neighbor method in high-dimensional spaces, which occurs when some vectors are nearest to many vectors while others are isolated.

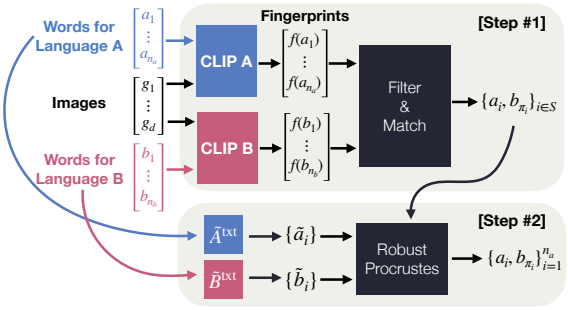


Figure 2: WALIP for translating between n_a words $\{a_1, \cdots, a_{n_a}\}$ and n_b words $\{b_1, \cdots, b_{n_b}\}$ in two languages A and B. We have access to: (1) a set of d images $\{g_i\}_{i=1}^d$, (2) the CLIP model for each language, and (3) static word embeddings for each language, denoted by $\tilde{A}^{\text{txt}}$ and $\tilde{B}^{\text{txt}}$. In step 1, we build a fingerprint $f(a_i)$ defined in equation 1 for each word a_i and build $f(b_i)$ for words b_i. We match words whose fingerprints share high similarities, thus having an initial mapping $\pi : [n_a] \to [n_b]$ pairing a_i and b_{π_i} for $i \in S \subseteq [n_a]$. In step 2, we use static word vectors and initially matched pairs to solve the linear word mapping with the robust Procrustes algorithm for better alignment.

4 WALIP

We first provide the high-level idea and then specify each stage of WALIP. Algo. 2 in Appendix presents the pseudocode for our algorithm.

4.1 Method Overview

Our idea is to enable effective and robust word alignment by using (1) the similarity of visual representations of words with similar meanings and (2) the structural similarity of static word embedding spaces across languages. Specifically, we use images to connect similar words in two languages with the aid of CLIP (Radford et al., 2021). However, a naïve application of this method only makes sense for visual words such as non-abstract nouns that images can describe. To map non-visual words,

we utilize the topological similarity (*i.e.*, the *degree of isomorphism*) between word vector spaces (Vulić et al., 2020). Motivated by the existence of a linear association between two static word embeddings of different languages (Ormazabal et al., 2019), we learn a linear mapping using the robust matching algorithm on identified word pairs.

Fig. 2 illustrates WALIP used for aligning words $\{a_i\}$ and words $\{b_i\}$ in languages A and B. WALIP has two steps. First, it selects pairs $\{a_i, b_{\pi_i}\}$ having similar visual meanings by using each word's fingerprint, defined as the similarity of the word and an image set via CLIP's encoders. Second, it iteratively aligns word embeddings of languages A and B, *i.e.*, find a linear mapping between two embeddings, using robust Procrustes and the initial pairs identified in the first step.

4.2 Step 1: Pairing up Visually Similar Words using Language-Image Association

As shown in Algo. 2, our Step 1 pairs words via images. This is available by an image-based fingerprint representation of each word, defined below.

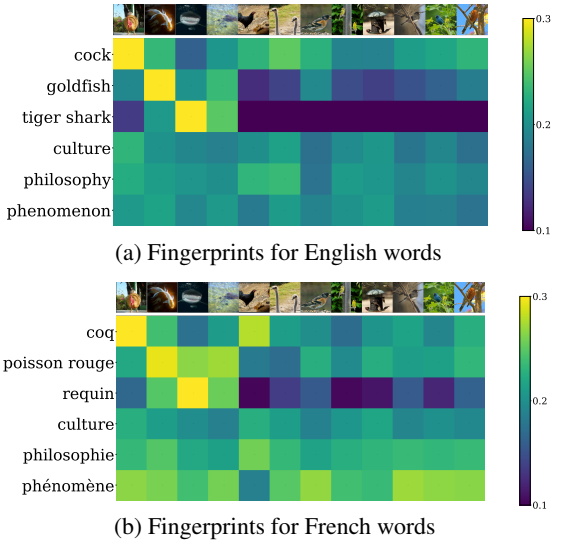
4.2.1 Image-based Fingerprints

We denote the image/text encoder of the CLIP model for language A as A^{img} and A^{txt}. Similarly, we define B^{img} and B^{txt} for language B. The critical advantage of the CLIP model is the access to the shared embedding space aligning image g_i and its corresponding word (a_i or b_i). WALIP utilizes this embedding space of each source/target language to find the bilingual mapping.

Given d images $\{g_1, \cdots, g_d\}$, we first define a $d-$dimensional vector (called **fingerprint**) for each word $a_i \in A_{\text{dict}}$ in the source language as $f(a_i) = [f^a_{i,1}, \cdots, f^a_{i,d}]$ where $f^a_{i,j} = \text{sim}(A^{\text{txt}}(a_i), A^{\text{img}}(g_j))$ is the similarity between the embedding of the i-th word and the embedding of the j-th image. Similarly, we define the fingerprint of each word $b_i \in B_{\text{dict}}$ in the target language as $f(b_i) = [f^b_{i,1}, \cdots, f^b_{i,d}]$ where $f^b_{i,j} = \text{sim}(B^{\text{txt}}(b_i), B^{\text{img}}(g_j))$. This fingerprint represents a word's similarity to images, according to the embedding space of pretrained CLIP models. We denote the fingerprint of the i-th word in the dictionary of a language $l \in \{a, b\}$ as

$$f(l_i) = [f^l_{i,1}, \cdots, f^l_{i,d}]. \quad (1)$$

Figs. 3a, 3b show examples of English and French fingerprints. Here, we measure the similarity of each word with 12 images from ImageNet (Deng et al., 2009), obtaining a 12-dim vector. The top three rows of each figure are fingerprints for visual words (`cock`, `goldfish`, `tiger shark`), and the bottom rows are of abstract words (`culture`, `philosophy`, `phenomenon`). Unlike visual words, fingerprints of abstract words are more uniform (similar values for most coordinates), *i.e.,* they are *not* distinguishable. Note that fingerprints of each English-French pair of visual words {(`cock`, `coq`), (`goldfish`, `poisson rouge`), (`tiger shark`, `requin`)} share highly similar patterns.



(a) Fingerprints for English words

(b) Fingerprints for French words

Figure 3: Illustration of image-based fingerprints for English words (a) and their translations in French (b). The similarity between each word (inserted in a simple template such as "`A photo of []`") and all images serves as the fingerprint (each row). Fingerprints of visual words (top three rows) are more distinguishable than abstract words (three bottom rows) and share similar patterns to the fingerprints of their French translations.

4.2.2 Identifying Pivot Pairs

Consider two visual words a_i, b_j in two languages with similar meanings (*e.g.*, $a_i =$ "`tiger shark`" and $b_j =$ "`requin`" in Fig. 3). For a given image set, fingerprints of the two words would be similar, *i.e.*, $f(a_i) \approx f(b_j)$ as shown in Fig. 3, allowing the use of fingerprint similarity for word translation.

Keeping only visually aligned words. Recall that fingerprints are meaningful for visual words only, as observed in Fig. 3. Motivated by this observation, we focus on words well represented by a set of images. Specifically, for the i-th word l_i in language $l \in \{a, b\}$, we compute the maximum similarity value $f^{(l)}_{i,\max} = \max_j f^{(l)}_{i,j}$ within the corresponding fingerprint $f(l_i) = [f^{(l)}_{i,1}, \cdots, f^{(l)}_{i,d}]$. Then, for each language $l \in \{a, b\}$, we keep the set of words S_l having the maximum similarity be-

yond the median. To focus on components with high similarity, we sparsify fingerprints by eliminating values below the 0.9^{th}-quantile and normalize the vectors. This revised fingerprint allows us to focus on images highly similar to the given word.

Selecting pairs with high similarity. For source words $\{a_i\}_{i \in S_a}$ and target words $\{b_j\}_{j \in S_b}$, we measure the similarity of fingerprints $f(a_i)$ and $f(b_j)$ using CSLS (Sec. 3). Recall that our goal is to find a mapping $\pi : [n_a] \to [n_b]$ indicating that the word a_i is translated to $b_{\pi(i)}$, and we want to map a_i to b_j having similar fingerprints. Based on the similarity score $c_{i,j} = \text{CSLS}(f(a_i), f(b_j))$ for $i \in S_a$ and $j \in S_b$, we set $\pi(i) = \arg\max_j c_{i,j}$, giving us an initial set of word pairs, where two words in each pair are visual words and share highly similar fingerprint patterns. See algorithms 3 and 4 for pseudocodes of word filtering and pair selection.

4.3 Step 2: Iteratively Learning the Mapping with Robust Procrustes

In Step 1 of WALIP in Algo. 2, we have identified the initial word mapping π on visual words. In Step 2, we learn and fine-tune π on the whole dictionaries using linear mapping $W^\star$ between *static* word embeddings of two languages, learned by iteratively applying our robust Procrustes algorithm (Algo. 1). We first explain Algo. 1 – the building block of Step 2 in Sec. 4.3.1, and then explain how this algorithm allows us to learn π in Sec. 4.3.2.

Algorithm 1 `Robust-Procrustes`

Input: Vectors $X, Y \in \mathbb{R}^{n \times d}$
Output: Linear mapping $W^* \in \mathbb{R}^{d \times d}$
Set $\epsilon = 0.001$, $M = 5$
Initial mapping $W_0 = \texttt{Procrustes}(X, Y)$
for $m \in \{1, \cdots, M\}$ **do**
  $\alpha_i \leftarrow \frac{1}{\|y_i - W_{k-1} x_i\|^2 + \epsilon}$ for $i \in [n]$
  $\alpha_i \leftarrow \alpha_i / \max_{j \in [n]} \alpha_j$
  $D \leftarrow \texttt{Diag}(\alpha_1^{1/2}, \ldots, \alpha_n^{1/2})$
  $W_m \leftarrow \texttt{Procrustes}(DX, DY)$
$W^\star \leftarrow W_M$

4.3.1 Error-Weighting Robust Procrustes

The initial word pairs identified in Step 1 are obtained in an unsupervised manner with potentially many mismatched pairs. Thus directly applying the existing Procrustes algorithm (Sec. 3) to these pairs may lead to an incorrect linear mapping W.

We introduce a robust matching algorithm (Algo. 1) to eliminate the mismatched pairs and learn the mapping from the correct ones. Inspired by the existing robust Procrustes algorithms (Groenen et al., 2005), we assign small weights to incorrect pairs and large weights to correct pairs. Given a word embedding matrix X and its aligned counterpart Y, we first apply the Procrustes to learn the initial W_0. We then measure the error of W_0 on each word pair (x, y) by the residual $r(x, y) = \|y - W_0 x\|_2$. Since the pair is likely to be correct when the residual is small, we use $\alpha(x, y) = 1/r(x, y)$ as the weight of the pair. Then, we apply Procrustes on these weighted pairs to obtain a new mapping W_1. We repeat this process a few times to achieve a stable linear mapping $W^\star$.

4.3.2 Iteratively Updating the Word Alignment π and Linear Mapping $W^\star$

In Step 2 of WALIP, we iteratively apply two procedures: first, we update linear mapping $W^\star$ by applying the robust Procrustes on identified pairs, and second, we update the word mapping π using $W^\star$ and the pair selection algorithm (Algo. 4).

The first phase is described in Sec. 4.3.1. In the second phase, we transform each source vector x_i into $W^\star x_i$ in the target embedding space and apply the k-nearest-neighbor (NN) on this space. We update π using Algo. 4 in the following manner: retrieving $k > 1$ candidate target words for each source word and choosing candidates having the similarity (with source word) higher than a threshold q. For the updated π, we measure the Euclidean distance between paired vectors as the validation loss and repeat the two procedures (update $W^\star$ and π) until the validation loss is convergent. In this process, two hyperparameters q and k are initialized with high values and gradually decayed at each update step of π. Once the validation loss converged, we obtain the final mapping π by applying Algo. 4 with $k = 1$ and $q = 0$.

Step 2 is crucial to achieving high translation performance from initial mapping. While sharing similar merits to ours, the refinement procedure (Conneau et al., 2017) is only used for marginally improving upon a high-accuracy linear mapping W.

4.4 Advantages of WALIP

First, WALIP is *computationally efficient*, especially compared to MUSE, MUVE, and GLOBETROTTER. With pretrained CLIPs, our first step (Sec. 4.2) requires no extra training for pivot pair matching, while Step 2 (Sec. 4.3) involves a few matrix computations. Second, WALIP is more *robust to language dissimilarity*. Assuming well-

trained CLIPs, fingerprints of words having similar meanings are intuitively similar across languages as they all represent the same visual correlation to the same image set. Thus, fingerprints improve the robustness of pivot matching, especially for dissimilar languages. This may not be the case for methods only using static word embeddings (Søgaard et al., 2018). Finally, our image-based fingerprint provides an *interpretable representation* of words.

5 Experiments

We evaluate WALIP on bilingual alignment tasks. Sec. 5.2 compares WALIP and baselines in multiple language pairs. The following sections provide additional experimental results that either highlight the benefits of WALIP or help understand the component that enables the high performance of WALIP. Our code is available at `https://github.com/UW-Madison-Lee-Lab/walip`.

5.1 Settings

WALIP setting. We use publicly available pretrained CLIPs for English, Russian, Korean, and Japanese. For other languages, we fine-tune English CLIP models on Multi30K (Elliott et al., 2016, 2017) and MS-COCO variants (Lin et al., 2014; Scaiella et al., 2019; Carlos, 2020). For making CLIP prompts, we convert single words to sentences using prompt templates suggested in (Radford et al., 2021). We apply the prompt-ensemble technique with 2–7 prompts for each word and use their average as word embeddings. To make the fingerprints, we use a set of 3000 images from ImageNet (Deng et al., 2009) by default. See Sec. 5.6.3 for our detailed evaluation. For the static word embedding, we use HowToWorld (HTW)-based Word2Vec (Sigurdsson et al., 2020) and Wiki-based Fasttext embeddings (Bojanowski et al., 2016).

Evaluation. We evaluate methods on the *Dictionary* datasets (Sigurdsson et al., 2020), which are test sets used in the MUSE benchmark (Conneau et al., 2017). Each dictionary is a set of translation pairs where each word in the source language may have multiple translations in the target language. We report recall@n used in (Sigurdsson et al., 2020), which presents the fraction of source words correctly translated. A retrieval is correct for a given query if at least one of n retrieved words is the correct translation. By default, we report recall@1, which is equivalent to precision@1, and the accuracy used in (Conneau et al., 2017).

Baselines. Our baselines include the video-grounding method *MUVE* (Sigurdsson et al., 2020) and the image-grounding method *Globetrotter* (Surís et al., 2020). We also compare our method with two versions of the text-only method *MUSE* (Conneau et al., 2017): the default one trained on the Dictionary dataset (with 1.5K–3K words per dictionary), and the other one trained on the MUSE training data (with 200K words per dictionary); we call the latter one as *MUSE (extra-vocabulary)*. We also consider a simple baseline using CLIP, denoted by *CLIP-NN*, which performs 1-nearest neighbor (1-NN) based estimation on the embedding spaces of two CLIP models: we first find the image nearest to the source word, and then find the target word nearest to the image found in the first step. For measuring recall@n of this baseline, we replace 1-NN with $\lceil\sqrt{n}\rceil$-NN.

We also test three variants of our method by making changes in Step 1: WALIP *(clip-text in Step 1)* which replaces fingerprints with CLIP-based text embeddings, WALIP *(substring matching)* which replaces the initial matching by selecting pairs sharing the longest common substrings, and WALIP *(character mapping)* which improves *substring matching* by first applying letter counting (Ycart, 2012) to map two languages' character sets. We also test two variants that replace the static word embeddings used in Step 2 with CLIP-based text embeddings (denoted by WALIP *(clip-text in Step 2)*) or fingerprints (denoted by WALIP *(fingerprint in Step 2)*). Further details are in Appendix B.

5.2 How Well Does WALIP Perform Bilingual Word Alignment?

Tables 1, 2 show our evaluation of bilingual alignment using Wiki-based and HTW-based embeddings on the Dictionary datasets.

Wiki-based embeddings. In Table 1, WALIP achieves comparable or the best performances in most cases among unsupervised methods, attaining relatively small gaps to the full supervision. Specifically, WALIP achieves SOTA on five pairs, especially for En→Ko, where WALIP outperforms others with large margins. For the baselines using visual information, we outperform GLOBETROTTER and all variants of WALIP across all pairs. Note that MUVE only reports recall@10 for En→Fr as 82.4, far below ours (97.5). Compared to the version of MUSE with extra vocabularies, WALIP achieves comparable scores in most cases and out-

Table 1: Comparing bilingual alignment methods on Wiki-based word embedding. We report recall@1 on the Dictionary dataset. WALIP achieves SOTA performance in many pairs, close to the supervision. *(Sigurdsson et al., 2020) do not report results of MUVE in this setting and* GLOBETROTTER *uses its learned word embeddings.*

	Method	En→Ko	En→Ru	En→Fr	En→It	En→Es	En→De	Es→De	It→Fr
Text-only	(Upper bound) Supervision	69.1	85.5	93.5	92.1	93.3	92.5	91.5	95.1
	MUSE (extra-vocabulary)	59.3	**83.0**	**92.5**	**91.6**	**93.0**	**92.5**	**89.1**	**94.5**
	MUSE	2.8	65.9	84.5	84.9	85.1	73.6	83.0	92.3
	WALIP (substring matching)	0.2	0.0	92.0	90.3	92.0	92.1	88.7	94.3
	WALIP (character mapping)	0.2	5.0	90.9	0.1	0.1	0.3	0.5	0.5
Text-Image	CLIP-NN	2.5	9.4	1.3	10.5	8.2	7.1	7.3	6.5
	GLOBETROTTER	0.1	4.0	52.3	50.1	46.4	46.8	38.3	49.3
	WALIP (clip-text in Step 1)	0.3	0.0	58.9	79.4	56.2	50.8	46.5	52.5
	WALIP (clip-text in Step 2)	0.2	15.7	59.3	59.1	59.1	52.3	46.8	52.1
	WALIP (fingerprint in Step 2)	0.2	0.5	31.3	39.0	32.6	31.3	34.7	43.3
	WALIP	**62.3**	82.7	**92.6**	90.7	92.2	**92.6**	**89.2**	**94.5**

Table 2: Comparing bilingual alignment methods on HTW-based embedding. WALIP achieves highest recall@n scores on Dictionary dataset across all pairs.

Method	En→Fr		En→Ko		En→Ja	
	R@1	R@10	R@1	R@10	R@1	R@10
(Up.) Sup.	57.9	80.1	41.8	72.1	41.1	68.3
MUSE (extra.)	26.3	42.3	11.8	23.9	11.6	23.5
MUSE	0.8	6.6	0.3	3.1	0.3	2.5
MUVE	28.9	45.7	17.7	33.4	15.1	31.2
WALIP (substr.)	35.5	56.0	0.0	0.2	0.3	2.1
WALIP	**35.6**	**56.2**	**20.2**	**42.4**	**19.6**	**41.0**

performs in En→Ko. The score gaps between the two methods are larger in Table 2, as described in the next paragraph. It is worth mentioning that this version of MUSE needs a large number of extra vocabularies for training while WALIP directly performs on the test dictionaries. Moreover, most baselines (except CLIP-NN) require intense training for aligning embedding spaces, while WALIP needs a few matrix computations.

HTW-based embeddings. Following (Sigurdsson et al., 2020), we test for three language pairs (En→{Fr, Ko, Ja}). Table 2 compares WALIP with MUVE and the baselines that perform well in Table 1. Results of MUSE (extra.) and MUVE are from (Sigurdsson et al., 2020). WALIP outperforms other unsupervised baselines with large margins, achieving the SOTA for all pairs, with the recall@1 gaps to the second-best method (MUVE) being $6.7, 2.8$, and 4.5 for En→{Fr, Ko, Ja}, respectively. WALIP also outperforms the substring matching variant on En→{Ko, Ja}.

The performance on dissimilar language pairs. For both embedding types, WALIP works relatively well regardless of the similarity of language pairs. In contrast, most baselines do not perform well on a few or all dissimilar pairs (En→{Ko, Ja, Ru}). We expect that the low performance of the *substring matching* method partly comes from the dissimilarity of alphabets in such pairs.

Table 3: Comparing methods when static word embeddings of source and target languages are trained on different corpora. We report recall@1 on En→Fr translation evaluated on Dictionary dataset. WALIP outperforms other baselines across two settings.

Method	Wiki-HTW	HTW-Wiki
MUSE (extra.)	0.3	0.3
MUSE	0.3	0.2
VecMap	0.1	0.1
MUVE	32.6	41.2
WALIP	**34.3**	**60.0**

5.3 Robustness against the Dissimilarity of Static Word Embeddings

Following (Sigurdsson et al., 2020), we evaluate WALIP when the static word embeddings of source and target languages come from different training corpora: Wiki and HTW corpora. We also compare with *VecMap* (Artetxe et al., 2017), the baseline used in the MUVE paper. Table 3 compares WALIP with MUSE variants, VecMap, and MUVE on En→Fr.[1] WALIP and MUVE are more robust to the dissimilarity of word embeddings than MUSE variants and VecMap. In addition, WALIP outperforms MUVE on both settings. For instance, on the Wiki-HTW setting, recall@1 of WALIP is 60% while that of MUVE is 41.2%.

5.4 Can We Reuse CLIP Models Trained on English Texts for Other Languages?

Large-scale language models exhibit the strong ability of cross-lingual zero-shot transfer (Hu et al., 2020). We investigate whether WALIP can utilize a CLIP model trained on English texts (English-CLIP) for other languages. Intuitively, this is probably doable when the other language uses the same alphabet (and the same tokenizer). Here, we use the English-CLIP model to obtain fingerprints for all languages, resulting in a new version of WALIP,

[1]MUVE only provides results for the English-French pair.

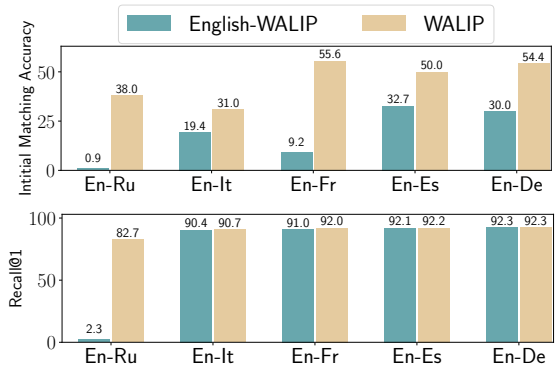


Figure 4: Zero-shot cross-lingual transfer. We observe the following when we replace the original CLIPs (yellow) with English-CLIP (cyan). Top: The initial matching accuracy drops for all pairs. Bottom: The final recall score becomes nearly 0 for the dissimilar pair (En→Ru) but remains mostly the same for other pairs.

Table 4: The percentage (%) of each word class in the Dictionary dictionaries. Each class of abstract and concrete nouns accounts for approximately 4% of words, with the total nouns being nearly 50% of words.

Dict.	Noun			Others
	Abstract	Concrete	Non-ID	
En→Ru	3.8	4.3	39.5	52.4
En→It	3.9	3.8	38.9	53.4

denoted *English*-WALIP. Here, our static word embeddings are Wiki-based Fasttext embeddings. As shown in Fig. 4, using English-WALIP causes drops in initial matching accuracies, which measure the precision of mapping on selected pairs. However, these drops only affect the translation performance of languages dissimilar to English (*e.g.*, Russian) and do not significantly affect the ones similar to English, *i.e.*, the recall@1 remains mostly the same for En→{It, Fr, Es, De}. Thus, English-CLIP can be used in WALIP framework for languages similar to English, reducing the need for training their new CLIP models.

5.5 WALIP on Different Word Types

In this section, we check how the performance of WALIP changes for different types of words. We categorize words into 4 classes: abstract nouns (*e.g.*, beauty), concrete nouns (*e.g.*, computer), non-identified nouns (*e.g.*, Copenhagen), and non-noun (*e.g.*, pretty). We use spaCy noun parser[2] to detect nouns and then use lists of popular English abstract and concrete nouns[3] to match their classes. We denote the unmatched nouns as non-identified (non-id). Table 4 reports the percentage of each class in the En→{Ru, It} dictionaries. Nearly 47% of words are nouns, with approximately 8% of words being abstract or concrete nouns.

[2]https://spacy.io

[3]englishvocabs.com/nouns/1000-concrete-and-abstract-nouns-examples, onlymyenglish.com/list-of-abstract-nouns

Table 5: Recall@1 (↑) of each word type reported on each step of WALIP. In the early stages, concrete nouns obtain the highest scores in both dictionaries. After step 2, abstract and concrete nouns share more comparable scores, higher than scores of non-noun words.

Dict.	Step (Iter.)	Noun			Others
		Abstract	Concrete	Non-ID	
En→Ru	#1	7.0	47.6	9.8	5.8
	#2 (first)	40.4	66.2	42.2	23.7
	#2 (last)	86.0	86.2	86.0	78.8
En→It	#1	3.3	35.1	13.2	12.9
	#2 (first)	72.9	77.2	68.0	55.1
	#2 (last)	96.6	94.8	92.0	89.3

Table 5 reports recall@1 scores for all word classes after the initial matching (Step 1 in Sec. 4.2) and after the first and the last iterations of linear mapping (Step 2 in Sec. 4.3). We use the Wiki-based Fasttext embeddings for static word embeddings. After completing step 1 and the first iteration of step 2, concrete nouns have the highest scores. Note that the score gap between concrete and abstract nouns on En→Ru is more than 40% after step 1. This indicates that the initial matching using fingerprints works better with concrete nouns. After completing step 2, the scores are improved for all classes, where scores of abstract and concrete nouns become more comparable, *e.g.*, 86.0, 86.2 on En→Ru. Note that nouns have much higher recall@1 than non-noun words. These results show that step 2 improves the matching for all word types, especially for nouns.

5.6 Ablation Study

We perform ablation studies using the Wiki-based Fasttext embedding and the Dictionary dataset.

5.6.1 Effect of Fingerprints

Fig. 5 shows the effect of fingerprints on translation performance. We compare variants of WALIP using various initial mapping methods: random matching (red), clip-text embeddings (olive), substring matching (green), and image-based fingerprints (ours, dark blue). The evaluation scores can be found in Table 1. Fingerprint-based WALIPs are the best among variants across all pairs.

5.6.2 Effect of Robust Procrustes

Fig. 6 shows the comparison between our robust Procrustes (in Algo. 1) and the standard Procrustes

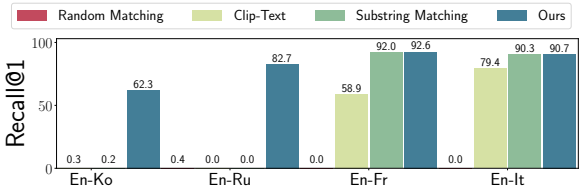


Figure 5: WALIP with different methods of initial mapping. Compared to image-based fingerprints (dark blue), using other methods for the initial mapping result in lower recall scores, especially for dissimilar language pairs (En→Ko and En→Ru).

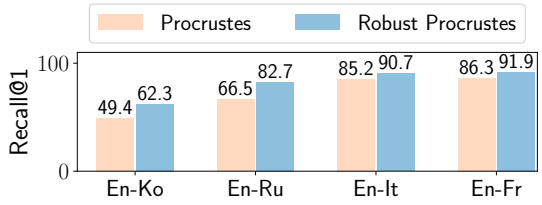


Figure 6: Investigating the effect of robust Procrustes. Robust Procrustes helps improve the translation across different language pairs. The effect is more significant on "difficult" pairs, such as English-Russian.

algorithm, given the same initial mapping. Robust Procrustes indeed helps improve over the standard Procrustes, especially when two languages are dissimilar. For instance, on En→Ko, using robust Procrustes increases the final recall@1 by 12.9%.

5.6.3 Effect of the Image Set

Here we check how the images used for making fingerprints affect the performance of WALIP.

Size of image sets. Fig. 7 compares recall@1 scores of WALIP when different number of images (from ImageNet) are used for building fingerprints. When the number of images increases, the recall@1 increases and converges for all pairs. As the languages become more dissimilar, WALIP may need more images to attain good performance. WALIP needs only 1000 to 3000 images to achieve good performance across all evaluated language pairs.

Diversity of images. To see the importance of image diversity, we fix the total number of images as 3000 and vary the number of classes. Table 6 compares the recall@1 of WALIP on En→Ru varying the image diversity. Here, we use the CIFAR10 dataset for 10 or fewer classes, CIFAR100 for 20–100 classes, and ImageNet for 1000 classes. Note that WALIP achieve high performance only when we use a large number of classes (e.g., more than 37 classes in the Table). This is probably because image sets with higher diversity provide more distinguishing coordinates of fingerprints to obtain more

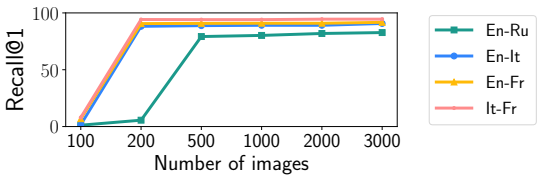


Figure 7: Recall@1 (↑) of WALIP varying the size of image (ImageNet) set used for fingerprints. The performance improves as the number of images increases from 100 to 1000 and then remains mostly unchanged. Hence, a sufficiently large number of images is required.

Table 6: Recall@1 (↑) of WALIP on En→Ru, varying the number of image classes given a fixed number of images as 3000. WALIP achieves high performance (step 2) when 38 or more classes are used. Furthermore, using 1000-class ImageNet results in the highest initial matching score (62.2) among the settings.

No. classes	Dataset	Step 1	Step 2
1	CIFAR10	0.9	0.8
2	CIFAR10	0.9	0.6
10	CIFAR10	6.2	8.1
20	CIFAR100	7.6	5.4
37	CIFAR100	9.1	4.4
38	CIFAR100	10.3	82.1
50	CIFAR100	11.1	82.5
100	CIFAR100	11.1	82.3
1 000	ImageNet	62.2	83.0

pivot pairs in the initial matching step – the condition for robust Procrustes to learn. Furthermore, compared to other settings, 1000-class ImageNet obtains much better initial matching in step 1.

6 Conclusion

We propose WALIP, a novel unsupervised bilingual word alignment method using pretrained CLIP models. WALIP first leverages the visual similarity between words as the auxiliary for matching initial and simple word pairs via the image-based fingerprint representation computed by language-image pretraining models. Then WALIP uses these initial pairs as pivots to learn the linear transformation between two static word embeddings. We introduce a robust Procrustes algorithm based on error-weighting to estimate the linear mapping. Compared with existing baselines, WALIP needs less computation for aligning two embeddings, thanks to the aid of visual information and pretrained CLIP models. WALIP achieves the SoTA alignment performances on several language pairs across word embedding types, especially for pairs in which two languages are highly dissimilar. WALIP also displays the robustness against the dissimilarity of static word embeddings' training corpora.

7 Limitations

Despite achieving high translation performance on various language pairs, WALIP has some limitations, coming from the requirements of CLIP models, the presence of visual words, and the structural similarity of static word embedding spaces.

As shown in Fig. 5, the initial mapping in Step 1 of WALIP needs to be sufficiently good for WALIP to achieve high translation performance. The conditions for good initial mappings are (1) well-trained CLIP models and (2) a sufficiently large number of visual words in the two dictionaries. *First*, our setting assumes the availability of pretrained CLIP models for the two languages. However, this may not be the case for many languages, especially for low-resource ones having small amounts of training data publicly available. We also observe that the CLIP models for non-English languages (either trained from scratch or fine-tuned from a model pretrained on English corpora) are not as good as the OpenAI CLIP trained on English corpora[4] in terms of image-text alignment and zero-shot image classification. Fortunately, our results on zero-shot transfer (Fig. 4) indicate that we may only need a few well-trained CLIP models in some major languages and further use them for their highly similar languages. *Second*, we have shown that image-based fingerprints work the best with visual words and may not show the distinguishable pattern on non-visual words (Fig. 3). Therefore, the two dictionaries need to have a sufficient number of visual words for WALIP to obtain initial pairs with adequate quantity and high accuracy.

Furthermore, WALIP, as well as most existing unsupervised word translation methods (Conneau et al., 2017; Artetxe et al., 2017; Sigurdsson et al., 2020) rely on the structural similarity of static word embedding spaces across languages. However, such linear mapping between two spaces may not exist in several cases, especially when two languages are highly dissimilar. For instance, we observed that the supervision method (with Procrustes) achieved low translation accuracy (approximately 40%) on the English-Japanese pair evaluated on the Dictionary dataset with HTW-based embeddings, indicating that the linear transformation assumption may not be fully satisfied for these two languages' static word embedding spaces.

[4] https://github.com/openai/CLIP

8 Broader Impact and Ethical Considerations

WALIP provides a simple yet effective solution to word translation, contributing to the progress of machine translation, which brings more benefits to our society. Our method is unsupervised and computationally efficient, thus significantly saving the computing and reducing the need for human labeling. Furthermore, the robustness of WALIP to the dissimilarity of language pairs and the dissimilarity of training corpora for static word embeddings may be beneficial to low-resource languages.

However, employing WALIP without careful consideration and understanding may lead to undesired outcomes. *First,* the provided dictionaries may contain harmful contexts and racist or sexist content. WALIP can be used to translate these contents to other languages, bringing unwanted adverse effects to society. *Second*, though achieving the SOTA performances, WALIP still has not attained sufficiently high accuracies (greater than 50%) on several dissimilar pairs (*e.g.*, En→Ja), potentially producing wrong translations for multiple words, and hence having undesired impacts to the users. *Third*, our methods may inherit biases and undesired contents from language-image (CLIP) models pretrained on large-scale datasets. Applying efficient fine-tuning to the pretrained CLIP models with fairness consideration methods (Gira et al., 2022) may help mitigate these biases.

References

Jean Alaux, Edouard Grave, Marco Cuturi, and Armand Joulin. 2018. Unsupervised hyper-alignment for multilingual word embeddings. In *International Conference on Learning Representations*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. 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)*, pages 451–462.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

García Carlos. 2020. MS-COCO-ES. *GitHub repository*.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Wietse de Vries and Malvina Nissim. 2020. As good as new. how to successfully recycle english gpt-2 to make models for other languages. *arXiv preprint arXiv:2012.05628*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 215–233, Copenhagen, Denmark. Association for Computational Linguistics.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74. Association for Computational Linguistics.

Michael Gira, Ruisu Zhang, and Kangwook Lee. 2022. Debiasing pre-trained language models via efficient fine-tuning. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 59–69, Dublin, Ireland. Association for Computational Linguistics.

John C Gower and Garmt B Dijksterhuis. 2004. *Procrustes problems*, volume 30. OUP Oxford.

Edouard Grave, Armand Joulin, and Quentin Berthet. 2019. Unsupervised alignment of embeddings with wasserstein procrustes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1880–1890. PMLR.

Patrick JF Groenen, Patrizia Giaquinto, and Henk AL Kiers. 2005. An improved majorization algorithm for robust procrustes analysis. In *New developments in classification and data analysis*, pages 151–158. Springer.

Mareike Hartmann, Yova Kementchedjhieva, and Anders Søgaard. 2019. Comparing unsupervised word translation methods step by step. *Advances in Neural Information Processing Systems*, 32.

John Hewitt, Daphne Ippolito, Brendan Callahan, Reno Kriz, Derry Tanti Wijaya, and Chris Callison-Burch. 2018. Learning translations via images with a massively multilingual image dataset. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2566–2576.

Yedid Hoshen and Lior Wolf. 2018. Non-adversarial unsupervised word translation. *arXiv preprint arXiv:1801.06126*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

Junjie Hu, Mengzhou Xia, Graham Neubig, and Jaime Carbonell. 2019. Domain adaptation of neural machine translation by lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2989–3001, Florence, Italy. Association for Computational Linguistics.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Jamie Kiros, William Chan, and Geoffrey Hinton. 2018. Illustrative language understanding: Large-scale visual grounding with image search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 922–933.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. 2022a. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *International Conference on Learning Representations (ICLR)*.

Yi Li, Rameswar Panda, Yoon Kim, Chun-Fu Richard Chen, Rogerio S Feris, David Cox, and Nuno Vasconcelos. 2022b. Valhalla: Visual hallucination for machine translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5216–5226.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640.

Rada Mihalcea and Chee Wee Leong. 2008. Toward communicating simple sentences using pictorial representations. *Machine translation*, 22(3):153–173.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

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

Aitor Ormazabal, Mikel Artetxe, Gorka Labaka, Aitor Soroa, and Eneko Agirre. 2019. Analyzing the limitations of cross-lingual word embedding mappings. *arXiv preprint arXiv:1906.05407*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(sept):2487–2531.

Antonio Scaiella, Danilo Croce, and Roberto Basili. 2019. Large scale datasets for image and video captioning in italian. *Italian Journal of Computational Linguistics*, 2(5):49–60.

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

Gunnar A Sigurdsson, Jean-Baptiste Alayrac, Aida Nematzadeh, Lucas Smaira, Mateusz Malinowski, Joao Carreira, Phil Blunsom, and Andrew Zisserman. 2020. Visual grounding in video for unsupervised word translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10850–10859.

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv preprint arXiv:1702.03859*.

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.

Dídac Surís, Dave Epstein, and Carl Vondrick. 2020. Globetrotter: Unsupervised multilingual translation from visual alignment. *arXiv preprint arXiv:2012.04631*.

Hagai Taitelbaum, Gal Chechik, and Jacob Goldberger. 2019. A multi-pairwise extension of Procrustes analysis for multilingual word translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3560–3565, Hong Kong, China. Association for Computational Linguistics.

Ivan Vulić, Sebastian Ruder, and Anders Søgaard. 2020. Are all good word vector spaces isomorphic? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3178–3192, Online. 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.

Ziyan Yang, Leticia Pinto-Alva, Franck Dernoncourt, and Vicente Ordonez. 2020. Using visual feature space as a pivot across languages. In *Findings of the*

Association for Computational Linguistics: EMNLP 2020, pages 3673–3678.

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. 2021. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*.

Bernard Ycart. 2012. Letter counting: a stem cell for cryptology, quantitative linguistics, and statistics. *arXiv preprint arXiv:1211.6847*.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. 2022. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017a. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1959–1970.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017b. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1934–1945.

Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2021. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Conditional prompt learning for vision-language models. In *The IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR)*.

Mingyang Zhou, Runxiang Cheng, Yong Jae Lee, and Zhou Yu. 2018. A visual attention grounding neural model for multimodal machine translation. *arXiv preprint arXiv:1808.08266*.

Appendix

Section A presents the pseudocodes for algorithms discussed in the main paper. We provide details of the experimental setting, chosen hyperparameters, computing resources, and running times in Section B for reproducibility.

A Algorithms

We present the pseudocodes for algorithms in Section 4 of the main paper, including the WALIP algorithm (Algo. 2), the visual-word filtering algorithm (Algo. 3), and the word matching algorithm (Algo. 4).

Algorithm 2 `WALIP`

Input: Source dictionary $A_{\text{dict}} = \{a_1, \cdots, a_{n_a}\}$, target dictionary $B_{\text{dict}} = \{b_1, \cdots, b_{n_b}\}$, CLIP models $(A^{\text{txt}}, A^{\text{img}})$, $(B^{\text{txt}}, B^{\text{img}})$, set of images $G = \{g_1, \cdots, g_d\}$, word vectors T_A for A_{dict}, T_B for B_{dict}, number of alignment steps M, threshold quantile q, number of candidates k.

Output : $\pi : [n_a] \to [n_b]$ such that $a_{\pi(i)} \equiv b_i$

`/* STEP 1. PAIRING USING FINGERPRINTS */`

for language $l \in \{a, b\}$ **do**

  $f(l_i) \leftarrow$ fingerprint in (1) for $i \in [n_l]$

$\mathcal{F} \leftarrow \{f(l_i)\}_{l \in \{a,b\}, i \in [n_l]}$

$\mathcal{F} \leftarrow$ `Visual-Word-Filtering`$(\mathcal{F})$

$\pi_0 \leftarrow$ `Matching-Filtering`$(\mathcal{F}, q)$

`/* STEP 2. ITERATIVE ROBUST PROCRUSTES */`

Set $Q_s = \{0.5, 0.5, 0.3, 0.1\}, K_s = \{10, 5, 3, 1\}$

Set $q = 0.5$, $k = 10$, $\epsilon_0 = \infty$, $\delta = 0.5$

for $m \in \{1, \cdots, M\}$ **do**

  $s^A_{m-1} \leftarrow \{i \in [n_a] : \pi_{m-1}(a_i) \in B_{\text{dict}}\}$

  $s^B_{m-1} \leftarrow \{j \in [n_b] : \exists a_i \text{ s.t. } \pi_{m-1}(a_i) = b_j\}$

  $T'_A \leftarrow T_A[s^A_{m-1}], \quad T'_B \leftarrow T_B[s^B_{m-1}]$

  $W^\star \leftarrow$ `Robust-Procrustes`(T'_A, T'_B)

  $T_A \leftarrow T_A W^\star$

  $\epsilon_m = \|T_A - T_B\|_F$

  if $\epsilon_m < \epsilon_{m-1} + \delta$ **then**

    $t \leftarrow \min\{\lceil M/10 \rceil, 4\}$

    $q \leftarrow Q_s[t], \quad k \leftarrow K_s[t]$

    $\pi_m \leftarrow$ `Matching-Filtering`$(\{T_A, T_B\}, q, k)$

$\pi \leftarrow$ `Matching-Filtering`$(\{T_A, T_B\}, 0, 1)$

B Experimental Setup, Implementation, and Running

We present details of the experimental setting (Sec. 5.1 in main paper) and the chosen hyperpa-

Algorithm 3 `Visual-Word-Filtering`

Input: Fingerprints $\mathcal{F} = \{f(l_i)\}_{l\in\{a,b\},i\in[n_l]}$
Output : Updated fingerprints $\mathcal{F}$
for $l \in \{a, b\}$ **do**
  $f^{(l)}_{i,j} \leftarrow j$-th element of $f(l_i)$, for $j \in [d]$
  $f^{(l)}_{i,\max} \leftarrow \max_j f^{(l)}_{i,j}$ for $i \in [n_l]$
  $S_l \leftarrow \{i : f^{(l)}_{i,\max} \geq \text{median}_i(f^{(l)}_{i,\max})\}$
  for $i \in S_l$ **do**
    $\bar{q} \leftarrow$ `0.9-th quantile of` $\{f^{(l)}_{i,k}\}^d_{k=1}$
    $f^{(l)}_{i,j} \leftarrow f^{(l)}_{i,j} \cdot \mathbf{1}_{\{f^{(l)}_{i,j}\geq\bar{q}\}}$
    $f^{(l)}_{i,j} \leftarrow f^{(l)}_{i,j}/|f^{(l)}_{i,j}|$

Algorithm 4 `Matching-Filtering`

Input: $\mathcal{F} = \{f(l_i)\}_{l\in\{a,b\},i\in[n_l]}$,
Threshold quantile q,
Number of candidates k ($k = 1$ by default).
Output : Word index mapping $\pi : [n_a] \to [n_b]$
$c_{i,j} \leftarrow \text{CSLS}(f(a_i), f(b_j))$ for $i \in [n_a], j \in [n_b]$
$\bar{c} \leftarrow$ `q-th quantile of` $\{c_{i,j}\}$
$\pi \leftarrow$ empty mapping from $[n_a]$ to $[n_b]$
for $i \in [n_a]$ **do**
  $J^\star \leftarrow \{j \in [n_b] : c_{i,j} \geq k\text{-th} \max_j c_{i,j}\}$
  $\pi(i) \leftarrow \{j \in J^\star : c_{i,j} \geq \bar{c}\}$

rameters in (B.1), the computing sources, running time, and validation performance in (B.2).

B.1 Experimental Setup

Static word embeddings. We use two embeddings: HowToWorld (HTW)-based Word2Vec (Miech et al., 2019; Sigurdsson et al., 2020) that trains Word2Vec (Mikolov et al., 2013b) on HTW video datasets and Wiki-based Fasttext (Bojanowski et al., 2016) that trains Fasttext on the Wikipedia corpus.

Evaluation benchmark and datasets. We use the *Dictionary* datasets (Sigurdsson et al., 2020) which are test sets of MUSE bilingual dictionaries (Conneau et al., 2017). Each test set provides a set of matched pairs in two languages where each word in the source language can have multiple translations in the target language. For instance, the En→Fr dictionary has 1500 unique English words and 2943 corresponding French words. All pairs used in our evaluation are En→{Fr, Ru, It, Ko, Ja}, and It→Fr. Input evaluation dictionaries are pre-processed to ensure the delimiting character is a white-space character and that there are no duplicate synonym pairs. Words that do not appear in the word2vec files for HowToWorld-based or Wiki-based embeddings were removed. We also provide the modifications of the original datasets that remove *non-native* words (*e.g.,* 'dot, gif' in the Korean dictionary). We provide all evaluated datasets in our source codes. The test dictionaries can also be found at `https://github.com/facebookresearch/MUSE` and `https://github.com/gsig/visual-grounding/tree/master/datasets`.

Evaluation metrics. Our metric is *recall@n* used in (Sigurdsson et al., 2020) for $n = 1, 10$: the retrieval for a query is correct if at least one of n retrieved words is the correct translation of the query. *Recall@n* presents the fraction of source words that are correctly translated. In our setting, the *recall@1* is equivalent to *precision@1*, and the matching accuracy used in (Conneau et al., 2017).

Baselines. We describe what baselines we have compared in this paper. **CLIP-NN** is a simple baseline that performs double 1-nearest neighbor (1-NN) on CLIP embeddings: Given a source word, we perform the 1-NN to find the nearest image (using source CLIP) and then perform the 1-NN on the target CLIP to find the nearest target word. For *recall@n*, we perform the similar double k-NN where $k = \lceil\sqrt{n}\rceil$. **MUSE** (Conneau et al., 2017) is a text-only method that learns the cross-lingual linear mapping via adversarially aligning embeddings' distributions and iterative refinement with Procrustes. As the adversarial training is sensitive to initialization, we follow the procedure in (Sigurdsson et al., 2020) and report the highest observed performance across different initializations on the test set. As a result, this represents an upper bound on the true performance of the baseline. **MUVE** (Sigurdsson et al., 2020) replaces the linear layer learned in the first stage of MUSE with the *AdaptLayer* learned by jointly training the embeddings of videos and captions, shared across languages. The *AdaptLayer* allows monolingual embeddings to be transformed into a shared space so the rest of the network can be shared, even if the input languages are different. Their results suggest that visually grounding translation with video allows for more robust translation. We use their reported performances (Sigurdsson et al., 2020) in our comparison. **Globetrot-**

ter (Surís et al., 2020) uses image-caption pairs to jointly align the text embeddings of multiple languages to image embeddings using a contrastive objective. Even though their model was trained on pairing sentences with images, they show that the text representation learned by their model can also be used for unsupervised word translation by using the Procrustes algorithm on the learned word embeddings. We use their word embeddings for word translation. We also evaluate the **supervision** method using the Procrustes on different ground-truth translation pairs and use its results as an upper bound of performance.

Implementation details. Here, we provide the details for implementing our algorithms.

CLIP models. We use publicly available pretrained CLIPs for English[5], Russian[6], Korean[7], and Japanese.[8] For other languages, we fine-tune English CLIP models on Multi30K (Elliott et al., 2016, 2017) and MS-COCO datasets (Lin et al., 2014; Scaiella et al., 2019; **?**) with translated captions for each target language. Precisely, we fine-tune each model for 20 epochs using the `NCEInfo` loss (Oord et al., 2018) without changing the architectures of the original CLIP's encoders. We use Adam optimizer (Kingma and Ba, 2014) ($\beta_1, \beta_2 = 0.9, 0.98$) with a learning rate of `1e-7` and cosine annealing scheduler (Loshchilov and Hutter, 2016).

Image datasets. We use 3000 images from ImageNet (Deng et al., 2009). We find that high-resolution images provide the best initial mappings among tested image data.

Prompts for words in CLIPs. As for the input of CLIPs, we convert every single word to a complete sentence. We use the prompt templates suggested in (Radford et al., 2021) and apply prompt-ensemble (Radford et al., 2021) for the best embedding. In particular, we use a set of (two to seven) prompts for each word and average these text embeddings as the word embedding.

Hyper-parameters. The robust Procrustes algorithm (Algo. 1) uses $M = 5$ iterations. In Algo. 2, we use $M = 40$ alignment iterations in Step 2 and select the best model by our evaluation loss. We observe that the evaluation losses on pairs of similar languages (*e.g.*, English-French) converge quickly after a few iterations, while the dissimilar pairs require more iterations. For quantile threshold q, we use the simple scheme by gradually reducing q from a set of discrete values $\{0.7, 0.5, 0.3, 0.1\}$. The number of candidates k is decayed using the following values $\{10, 5, 3, 1\}$.

[5] https://github.com/openai/CLIP

[6] https://github.com/sberbank-ai/ru-clip

[7] https://github.com/jaketae/koclip

[8] https://huggingface.co/rinna/japanese-clip-vit-b-16

Table 7: Estimated WALIP validation loss (Euclidean distance) on several language pairs performed on the HTW-based embedding and Dictionary dataset.

	En→Fr	En→Ko	En→Ja
Avg. Dist.	8.49	8.58	8.55

Table 8: Estimated WALIP validation loss (Euclidean distance) on several language pairs performed on the Wiki-based embedding and Dictionary dataset.

	En→Ko	En→Ru	En→Fr	En→It
Avg. Dist.	15.70	14.24	10.99	11.67
	En→Es	En→De	Es→De	It→Fr
Avg. Dist.	10.79	12.06	13.28	11.54

B.2 Computation and Evaluation of WALIP

Validation. As WALIP is unsupervised, we estimated the validation error (or loss) by evaluating the average squared Euclidean distance between the mapped source embeddings and the target word embeddings. We use this criterion to select our best mappings. We report the validation errors in Table 7 and Table 8 for evaluated language pairs on two types of static word embeddings. We can see that the validation errors of the dissimilarity of language pairs (*e.g.,* En→Ko) are higher than the similar pairs (*e.g.*, En→Fr). We report the *recall@1* scores corresponding to mappings with the smallest validation errors.

Computing resources and time. We run our algorithms and baselines on an NVIDIA GeForce RTX 3090 GPU. The average running time of WALIP is about less than 2 minutes, while MUSE models take approximately an hour for each language pair.

Number of parameters of WALIP models. Each of our pretrained CLIP models has about 150 million trainable parameters. In ablation studies, we have tested WALIP with larger versions of CLIP models, with upwards of 400 million trainable parameters. However, we find that both WALIP versions with smaller and large CLIP models share similar translation performances across different language pairs.